컨테이너 환경에서의 RDMA NIC 마이크로아키텍처 자원 고갈 영향 분석*

김건우[1], 김진우[2,†], 박병준[2]
[1,2]광운대학교 (대학원생, 교수)

Analyzing the Impact of RDMA NIC Microarchitecture Resource Exhaustion in Containerized Environments*

Gun-Woo Kim[1], Jin-Woo Kim[2,†], Byung-Joon park[2]
[1,2]Kwangwoon University(Graduate Student, Professor)

요 약

최근 컨테이너화된 클라우드 환경에서 CPU 오버헤드를 줄이고 고성능 데이터 교환을 위해 RDMA (Remote Direct Memory Access)의 도입이 확산되고 있다. 이를 위해서는 한 컨테이너의 RDMA 워크로드가 다른 컨테이너의 RDMA 성능에 부정적인 영향을 미치지 않도록 높은 수준의 성능 격리가 필수적이다. 그러나 기존의 성능 격리 기법은 RDMA NIC (RNIC)의 복잡한 마이크로아키텍처 자원 관리 문제로 인해 효과적으로 적용되기 어렵다. 본 논문에서는 컨테이너 환경에서의 악의적인 RDMA Verbs가 RoCEv2 기반 BlueField-3 RNIC의 마이크로아키텍처 자원에 미치는 영향을 실험적으로 분석하였다. 공격 시나리오를 설정하고 실험한 결과, 대역폭은 약 93.9% 감소, 지연 시간은 약 1,117배 증가하였으며 캐시 미스도 115% 증가함을 확인하였다. 공격 시나리오에 대한 구체적인 분석을 기반으로 Threshold 기반 자원 관리 방식인 HT-verbs를 제안하고 RNIC의 문제점을 완화할 수 있는 방안을 제시한다.

I. 서론

RDMA (Remote Direct Memory Access)는 원격 프로세스가 RDMA NIC (RNIC)을 통해 호스트 메모리에 커널을 우회하여 직접 접근할 수 있게 하는 기술이다. Host CPU의 소모를 적게하고 빠른 데이터 교환을 가능하게 하여 최근 클라우드 환경에서 널리 도입되고 있다. 또한 RoCEv2 (RDMA over Converged Ethernet version 2)는 기존 RoCE v1이 L2 서브넷 내에서만 통신할 수 있다는 문제점을 해결하여 RDMA 기술의 배포를 촉진시켰다.

RDMA는 이렇듯 클라우드 환경에 여러 장점을 부여하였지만 한가지 문제점이 존재한다. 대표적인 것은 이른바 성능 격리(performance isolation)로 알려진 문제이다[1]. 이는 악의적인 의도를 가진 컨테이너가 특정 RDMA 연산(예: control verbs, data verbs, exception)을 수행할 때, RDMA의 마이크로아키텍처 자원 (microarchitecture resource)이 고갈되어 다른 테넌트의 성능을 크게 저하시키는 공격을 말한다[2]. 또한 캐시 미스, 자원 고갈에 따른 서비스 거부 문제도 발생할 수 있다.

이와 같은 RDMA의 성능 격리 문제는 여러 이전 연구에서 다루어졌다[3]. 그러나 이들은 대부분 가상머신 또는 베어메탈 환경에서 분석되었다는 한계점이 있다. 최근 클라우드의 동향은 컨테이너(container)를 사용한 클라우드 네이티브 아키텍처를 지향하기 때문에 RDMA 또한 컨테이너에 맞게 도입하려는 움직임을 보이고 있다. 예를들어 대표적인 RNIC 벤더인 NVIDIA 역시 쿠버네티스에 호환되는 RDMA 플러그인을 제공한 바 있다. 그러나 컨테이너 환경에서 RNIC의 마이크로아키텍처 자원이 고갈되었을 때의 영향은 분석된 바가 없다. 향후 RDMA와 컨테이너 둘 모두를 활용한 배포 시나리오가 증가할 것을 고려할 때 이를 사전에 분석해보는 것은 매우 중요한 과제이다.

* 본 연구는 환경부의 통합환경관리특성화대학원 사업의 지원을 받았습니다.
† 교신저자(jinwookim@kw.ac.kr)

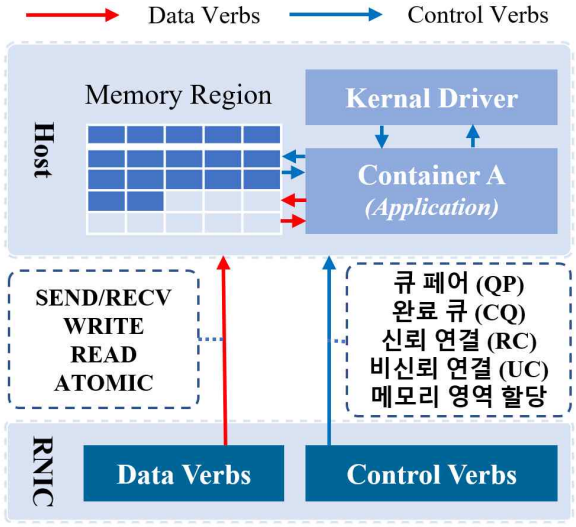

Fig. 1 Overview of RDMA workflow

본 논문에서는 공격자 컨테이너가 RNIC 장비인 NVIDIA BlueField-3의 마이크로아키텍처 자원을 고갈시킬 때 정상 컨테이너 성능에 미치는 영향을 분석한다. 실험 결과 희생자 컨테이너의 대역폭이 약 93.9% 감소하고 지연 시간은 약 1,117배 증가한 것을 보였다. 마지막으로 이를 해결하기 위한 Threshold 기반 접근 방식인 HT-verbs 을 제시한다.

II. 배경 지식

2.1 Remote Direct Memory Access

Fig. 1은 RDMA의 전체 워크플로우를 보여준다. RDMA는 Verbs라는 API를 통해 통신을 수행하며 Control Verbs와 Data Verbs로 구성되어있다. 컨테이너의 애플리케이션은 큐 페어(QP)와 완료 큐(CQ)와 같은 필요한 객체를 생성하고 신뢰 및 비신뢰 연결을 설정한다. 이후 호스트의 DRAM 영역을 할당하고 가상 주소에서 물리적 주소로의 매핑을 수행하여 RNIC이 커널 간섭 없이 메모리 영역을 직접 읽거나 쓸 수 있도록 한다. 이러한 초기화 과정이 완료되면 애플리케이션은 로컬 및 원격 메모리 간 데이터 전송을 시작할 준비가 된다.

이후 해당 컨테이너를 사용하는 송신자는 SEND/RECV, WRITE, READ, ATOMIC 작업을 RDMA를 통해 커널을 우회하여 수행할 수 있다. 요청 유형에 따라 RNIC의 마이크로 아키텍처 자원에 미치는 영향이 달라진다.

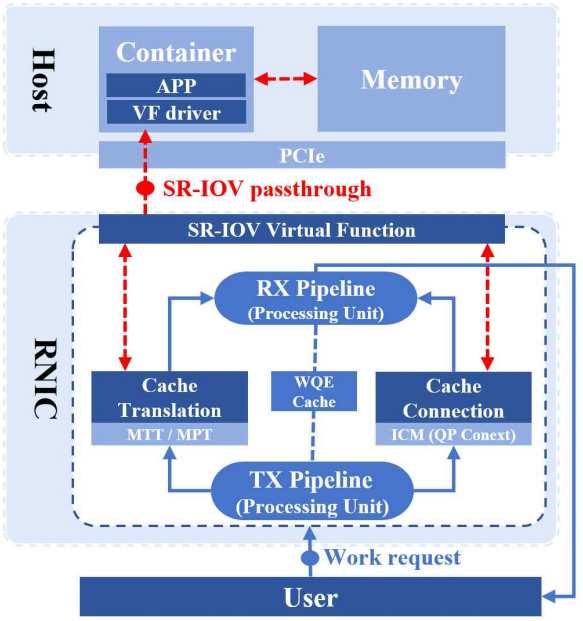

Fig. 2 Overview of RNIC microarchitecture resource flow

2.2 마이크로아키텍처 자원

Fig. 2는 컨테이너 환경에서의 RNIC을 통한 SR-IOV (Single Root I/O Virtualization) 기반 RDMA 통신의 마이크로아키텍처 자원 흐름을 나타낸다. SR-IOV를 통해 생성된 가상 함수(Virtual Function, VF)를 이용하여, 각 컨테이너는 RNIC의 자원에 접근하여 고성능 RDMA 통신을 수행할 수 있다. 이 과정에서 다양한 마이크로아키텍처 자원이 사용된다.

RDMA 작업 중 NIC 내부에는 여러 유형의 캐시와 프로세싱 유닛(PU)이 동작하며, 각 캐시는 특정 메타데이터 저장 및 빠른 접근을 지원한다. BlueField-3를 예시로 들면, Translate Cache는 MMT (Memory Translation Table)와 MPT (Memory Protection Table) 정보를 포함하며, 가상 주소를 물리적 주소로 변환하고, 메모리 접근 권한을 제어한다. 이를 통해 컨테이너의 RDMA 요청이 호스트의 물리 메모리에 안전하게 접근할 수 있도록 한다. Connection Cache는 ICM (Interconnect Connect Memory) 캐시로서 QP 컨텍스트(예: QP 상태 및 연결 정

보)를 저장하여 RDMA 작업의 연결 관리를 담당한다. 또한 WQE (Work Queue Entry) 캐시는 송신/수신 큐 항목을 미리 저장 하여 TX/RX 파이프라인이 빠르게 작업을 처리할 수 있도록 돕는다. 송신 요청은 TX 파이프라인을 통해 사용자 측으로 전달되며, 수신 요청은 RX 파이프라인을 통해 컨테이너의 메모리로 전달된다.

이와 같이 컨테이너화된 클라우드 환경에서는 여러 컨테이너가 하나의 RNIC을 공유할 수 있다. 그러나 클라우드 제공자가 컨테이너 내부 동작이나 자원 사용을 일일히 감시하기 어렵기 때문에 RNIC의 마이크로아키텍처 자원이 악의적으로 사용될 수 있다. 따라서 RDMA Verb가 소비하는 마이크로아키텍처 자원을 이해하고 적절히 할당하여 여러 컨테이너가 예측 가능한 성능을 유지 할 수 있도록 해야 한다.

III. 마이크로아키텍처 자원 고갈 공격

3.1 위협 모델

공격자는 특정 Control Verbs 및 Data Verbs 남용을 통해 과도한 큐 페어, 완료 큐 생성을 유발하고 RNIC의 마이크로아키텍처 자원을 고갈 시킬 수 있다고 가정한다. 또한 공격자는 의도적으로 예외 상황을 유발시켜 RNIC의 예외 처리 유닛을 과부하 시킴으로써 다른 컨테이너의 RDMA를 서비스 불능으로 만들 수 있다.

3.2 공격 시나리오

본 연구에서는 두가지 시나리오를 제시한다.

1) 큐 플러딩 공격: 다수의 큐 페어(QP)를 생성하거나 생성한 완료 큐(CQ)를 통해 지속적으로 데이터를 송수신함으로써 RNIC에 과도한 부하를 유발하는 방식이다. 이는 TX/RX PU의 처리 능력을 초과시켜 정상적인 데이터 전송 작업의 지연을 유발함으로써 전체 시스템의 대역폭을 감소시킨다.

2) 캐시 소진 공격: 공격자가 지속적으로 새로운 메모리 위치에 접근하는 RDMA 연산을 수행하여 캐시 히트율을 낮추고 Cache Translation을 처리하는 캐시를 고갈 시키는 방식이다. 이는 해당 캐시 자원를 소모시켜 정상적인 RDMA의 캐시 미스 비율이 증가한다.

두 시나리오는 마이크로아키텍처 자원을 과부하하여 고갈시키기 흐름을 확인하기 위해 부하를 점진적으로 증가시키는 방식이다.

3.3 실험 환경

실험은 BlueField-3 RNIC과 SR-IOV를 활용하여 공격자와 희생자 컨테이너를 도커 기반으로 격리된 환경을 구축하였다. 각 컨테이너는 독립적인 VF 드라이버를 통해 SR-IOV로 생성된 VF에 직접 접근할 수 있으며, 독립된 큐 페어(QP)와 완료 큐(CQ)를 사용하도록 설정하여 서로 다른 컨테이너 간의 성능 격리를 시뮬레이션 환경을 구축하였다. 또한 RoCEv2 프로토콜을 기반으로 RDMA 통신을 설정하였다.

테스트 환경에서는 RDMA 성능 벤치마킹 도구인 perftest를 사용하여 RDMA 대역폭을 측정하였다(예: ib_write_bw, ib_read_lat). 이를 통해 공격 전후의 대역폭 변화를 관찰 하였다. 캐시 고갈 공격의 영향을 정략적으로 평가하기 위해 perf를 사용하였다.

3.4 공격 영향 분석

Fig. 3는 첫 번째 시나리오의 결과로, TX/RX PU 부하에 따른 영향을 나타낸다. 공격자가 5초 마다 QP 및 CQ 자원을 과도하게 사용함에 따라, 희생자의 대역폭이 26.61 Gbit/sec에서 1.64 Gbit/sec으로 약 93.9% 급격히 감소한 것을 확인할 수 있다. 이는 TX/RX 처리 유닛이 점진적으로 과부하되면서 희생자 컨테이너의 대역폭이 크게 저하되었음을 의미한다.

Fig. 4는 두 번째 시나리오 결과로, RDMA 연산에서 새로운 메모리 위치에 지속적으로 접근하여 캐시를 고갈시키는 공격의 영향을 보여준다. 초기에는 지연 시간이 1.56μs로 RDMA의 장점을 확인할 수 있었으나, 시간이 지남에 따라 1,746.34μs로 약 1,117배(111,700%) 급격히 증가하는 것을 확인할 수 있다. 캐시 미스 비율 또한 14.48%에서 31.07%로 약 115% 증가하였다. 이는 캐시 자원의 고갈로 인해 RDMA 통신의 성능이 저하되었음을 나타낸다.

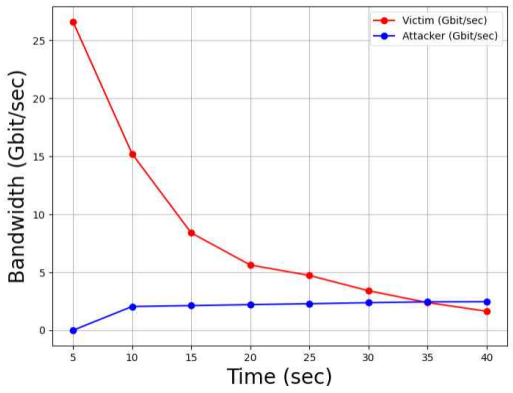

Fig. 3 Bandwidth changes due to TX/RX processing unit overload

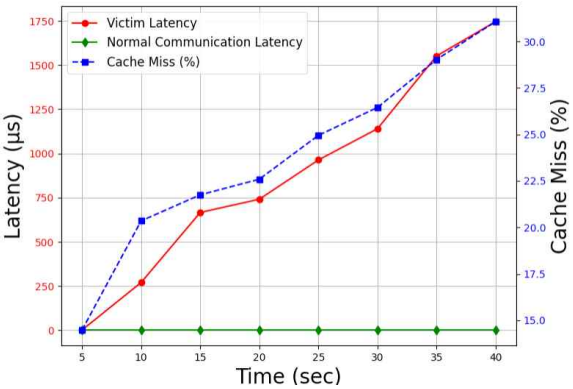

Fig. 4 Latency and cache miss change rates due to resource depletion

두 공격 시나리오는 NIC의 마이크로아키텍처 자원을 고갈시켜 그 영향도를 보여주었다. 이러한 큰 영향에도 불구하고 RDMA는 호스트의 CPU,메모리를 직접 우회하여 NIC 자원에 접근하기 때문에 기존의 일반적인 모니터링 방식으로는 공격을 효과적으로 탐지하는데 한계가 있다.

3.5 HT-verbs: Threshold 기반 자원 관리

RoCEv2기반 RNIC환경에서 발생할 수 있는 자원 고갈 공격을 완화하기 위해 'Threshold 기반 자원 관리' 방식을 제시한다. 특히 컨테이너 우선순위를 설정하고 특정 Verbs 및 캐시를 Hot, Warm, Cold로 분류하여 RNIC의 마이크로아키텍처 자원 과부화 문제를 완화한다.

분류하기에 앞서 각 컨테이너가 사용하는 RDMA Verbs (READ, WRITE, SEND, RECV, ATOMIC 등)의 호출 기록을 주기적으로 집계하여 특정 캐시 접근과 자원 소모율에 대한 패턴을 분석한다. 과거 및 현재 사용 패턴 변화를 감지하여 분류 기준을 업데이트한다. 분류 기준으로 RDMA Verbs를 Hot, Warm, Cold로 분류하며 주기적으로 상황에 맞는 동작을 수행한다. 특정 Verbs가 Hot인 경우 우선 순위가 낮은 컨테이너 접근 시 지연을 유도하거나 제한한다. Warm인 경우 자원 접근 시 동일한 격리를 제공한다. Cold시 기존 격리 환경보다 높은 성능을 제공한다. 이를 통해 다양한 상황에 따라 유연한 자원 대응을 가능하게 될 것이다.

IV. 결론

본 논문에서는 RoCEv2 기반 BlueField-3 RNIC의 성능 격리 문제를 실험적으로 분석하였다. 특히, RNIC의 마이크로아키텍처 자원인 TX/RX Processing Unit과 내부 캐시 자원이 고갈됨에 따라 컨테이너의 대역폭 감소, 캐시 미스율 증가, 그리고 지연 시간 악화 현상을 확인하였다. 이를 해결하기 위해 Threshold 기반의 자원 관리 기법을 제시하였다. 이 기법은 RDMA Verbs의 호출 패턴을 기반으로 마이크로아키텍처 자원을 Hot, Warm, Cold로 분류하고, 상황에 맞게 자원 할당을 조율하는 방식이다. 이를 통해 악의적인 컨테이너로 인해 발생할 수 있는 성능 저하 문제를 효과적으로 예방할 수 있을 것으로 기대된다.

향후 연구에서는 RNIC의 마이크로아키텍처에 영향을 미치는 RDMA Verbs 관점에서 분석하고, HT-verbs 시스템을 구축하여 실제 클라우드 환경에서 적용 가능성을 검증할 필요가 있다. 이를 통해 RNIC기반 클라우드 서비스의 성능 안정성을 한층 더 개선할 수 있을 것이다.

[참고문헌]

[1] Grant, Stewart, et al. "SmartNIC Performance Isolation with FairNIC: Programmable Networking for the Cloud", ACM, July, 2020

[2] Lou, Jiaqi, et al. "Harmonic: Hardware-assisted RDMA Performance Isolation for Public Clouds." 21st USENIX Symposium. 2024.

[3] Kong, Xinhao, et al. "Understanding RDMA microarchitecture resources for performance isolation." 20st USENIX Symposium. 2023.