반 가상화 환경에서 소켓버퍼 재사용을 통한 라우팅 성능향상

김도중○ 이신형 유 혁
고려대학교 컴퓨터·통신공학과
{djkim, shlee, hyuckyoo}@os.korea.ac.kr

Routing Performance Enhancement by Reusing Socket Buffer in Para-virtualized Environment

Dojung Kim○ Shin-Hyoung Lee Chuck Yoo
Department of Computer Science and Communication Engineering, Korea Univ.

요 약

최근 Xen의 반가상화 기법을 이용한 플랫폼이 각광받고 있다. 반가상화 기법은 게스트 도메인의 운영체제를 수정해야 하는 반면 성능 오버헤드를 최소화 하였다는 점에서 그 장점이 두드러진다. 그러나 가상화 플랫폼이 가지는 관리 계층의 호출에 따른 오버헤드는 여전히 존재한다. 특히 네트워크 패킷 송수신에 있어서의 관리 계층 호출에 대한 오버헤드는 전체 성능을 저하시킬 정도로 크다. 본 논문에서는 이러한 호출 스택이 가지는 문제점에 대해서 지적하고 소프트웨어 라우터에서 이를 극복하기 위한 네트워크 송수신 버퍼 할당 구조를 제안한다.

1. 서 론

최근 클라우드 컴퓨팅이 각광을 받으면서 가상화에 대한 논의가 진행되고 있다. 가상화는 물리적인 컴퓨팅 자원을 논리적인 여러개의 자원으로 독립시켜 이를 사용하는 독립적인 도메인을 구성하는 기술이다. 물리적인 하나의 자원을 나누어 쓰기 위해서는 각 도메인에서의 접근을 분리해주고 독립적으로 서비스를 해주어야 한다. 이러한 서비스를 제공해 주기 위해서 가상화 플랫폼들은 이를 관리하는 관리 계층을 별도로 두어 논리적으로 나뉜 자원에 대해 할당/해제 및 멀티플렉싱/디-멀티플렉싱.을 수행하고 있다. Xen[1] 에서는 이러한 서비스를 Hypervisor 라는 계층을 통하여 제공하고 있다. 이와 같이 자원을 관리하는 하나의 계층을 거쳐야하는 특성으로 인해 가상화된 환경에서는 일반적인 가상화 되지 않은 환경과 비교하여 오버헤드가 존재한다[2][3].

특히 네트워크 패킷을 수신하기 위해 커널 내에서는 버퍼를 계속적으로 할당 및 해제를 수행하고 이에 따라 DMA를 위한 메모리 매핑과정이 후속된다. 가상화된 도메인에서는 이들 매핑 요청에 대한 처리를 위하여 Hypervisor를 호출한다. 초당 수십-수백만개의 패킷이 수신되는 라우팅 환경에서는 지속적인 Hypervisor 호출이 오버헤드로 작용한다.

이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2010-0029180).

본 논문에서는 네트워크 패킷 수신을 위해 지속적으로 Hypervisor를 호출하는 과정의 오버헤드를 최소화 하고자 한다. 이를 위해 Xen을 이용한 반가상화 도메인을 이용하여 네트워킹 성능을 측정하고 이 과정에서 네트워크 버퍼 할당과정에서 Hypervisor 호출이 야기하는 성능 저하에 대해 분석하고 이를 극복하는 구조를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 Xen 에서의 DMA 매핑 과정에 대해 설명한다. 3장에서는 네트워크 패킷 송수신에서 DMA 매핑으로 인해 발생하는 오버헤드에 대해서 분석하고 이 과정이 성능저하를 발생시킴을 규명한다. 4장에서는 분석한 결과를 토대로 이를 극복하는 구조를 제안하고 5장에서 이러한 구조의 발전 방향을 제시하며 끝맺음을 한다.

2. Xen 가상화 환경

Xen은 반 가상화 기반의 Hypervisor로서 게스트 도메인의 운영체제에 수정이 필요하지만 성능에 대한 오버헤드가 적은 가상화 플랫폼이다. I/O 장치의 가상화는 Front-end와 Back-end로 구성되어 있다. Back-end는 관리 도메인에서 실제 장치를 사용하여 데이터를 처리는 드라이버이다. Front-end는 가상화 된 게스트 도메인에서 이들 장치를 가지고 있는 것처럼 보이게 하여 이곳으로 I/O를 요청하면 Front-end 에서는 이를 링 버퍼를 통해 Back-end로 다시 장치 사용을 요청하는 형태로 되어있다[4].

2-1. 반 가상화 환경에서의 DMA 매핑 과정

DMA 매핑을 위해서는 할당 받은 메모리 공간의 물리 메모리 주소를 장치로 알려주어야 한다. 또한 이 메모리 주소에 대한 사용 권한을 디바이스로 할당함으로써 다른 곳에서의 접근을 차단한다. 반 가상화 환경의 Xen에서는 이와 같은 과정을 Hypervisor로 요청을 하여 수행하게 된다(그림 1).

할당 받은 메모리의 가상주소로 물리주소를 알아내고 디바이스로 권한을 설정하기 위해 dma_map_single을 호출한다. 일반적인 경우 이 함수에서는 가상주소를 물리 주소로 변환하고 DMA가 가능한 하위 메모리로 Bounce Buffer를 만들어 해당 위치에 장치 접근 권한을 할당하게 된다. 그러나 Xen 게스트 도메인에서는 받은 가상 주소를 페이지 주소로 변환하고 다시 게스트 도메인이 바라보는 Pseudo Physical Memory 주소로 사상하여 이 주소를 가지고 Hypervisor로 요청을 보낸다. Hypervisor에서는 소프트웨어 I/O TLB에 디바이스가 접근 가능한 주소에 Bounce Buffer를 만들고 받아온 주소를 다시 실제 물리 메모리 주소로 변환하여 이 주소를 Bounce Buffer의 실제 주소로 등록하고 Bounce Buffer의 주소를 반환한다. 이와 같이 Hypervisor 호출 및 주소 변환 과정에 있어서 일반 환경에 비해 잦은 Context Switch로 인해 오버헤드가 발생하게 된다.

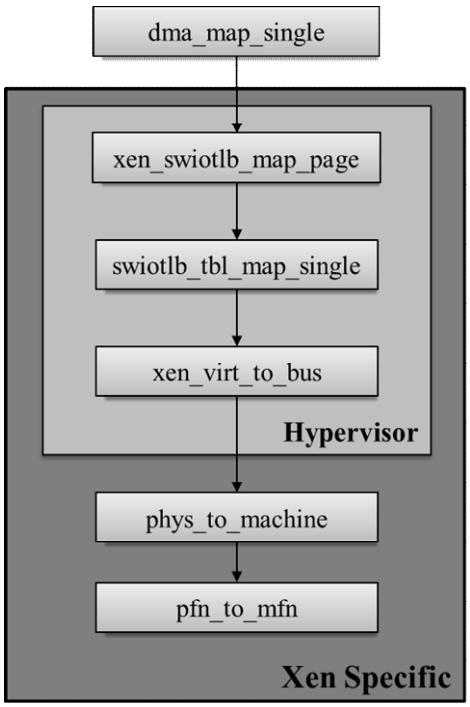

그림 1. Xen에서 DMA를 통한 물리 주소 획득 과정

3. 패킷 송수신을 위한 소켓버퍼 할당

패킷 송수신을 위해 리눅스 커널에서는 NIC 드라이버에서 소켓버퍼를 관리한다. 드라이버 내에서 소켓버퍼 관리 구조를 분석하기 위해 본 논문에서는 Intel 10Gb Ethernet NIC의 게스트 도메인 드라이버인 ixgbevf 2.0 버전을 이용하였다.

디바이스 드라이버에서는 최초 512개의 소켓 버퍼 메모리 풀을 링 형태로 만들고(Rx Ring) 수신 인터럽트를 받고 NIC의 Rx 버퍼를 4096번 폴링하여 최대 4096개의 패킷을 수신하면서 소켓 버퍼를 소모하게 된다. 소모된 소켓 버퍼는 매 16개의 소켓 버퍼 마다 새로 메모리를 할당 받고[1] Bounce Buffer 할당 및 버퍼 접근 권한 설정을 수행하게 된다.

송신과정에서는 수신과는 반대의 과정으로 송신이 필요한 소켓 버퍼 구조체를 받아와서 데이터 부분에 대한 Bounce Buffer 할당 및 버퍼 접근 권한을 설정한 후에 이를 Tx Ring에 저장시켜 DMA를 통해 디바이스로 데이터가 전송되고 전송이 완료되면 디바이스가 인터럽트를 통해 드라이버가 이 공간에 대해 메모리 해제를 수행한다.

4. 오버헤드 최소화를 위한 구조

소켓 버퍼를 소모할 때마다 새로이 DMA 매핑을 수행할 경우 이에 따른 Hypervisor 호출 오버헤드와 주소 변환 오버헤드가 상당히 크다. 특히 네트워크 패킷 입출력이 빈번한 소프트웨어 라우터 구조에서는 매번 이와 같은 매핑을 수행하였을 때의 생기는 오버헤드는 전체 성능을 좌우할 정도로 크다.

라우터의 특성상 한번 수신된 패킷은 다시 송신이 될 가능성이 크다. 또한 송수신 공히 DMA를 사용하므로 수신시에 할당 받은 DMA 물리 메모리를 해제하고 다시 송신 시에 물리 메모리를 예약 받는 과정이 사실상 중복되는 과정이라 할 수 있다. 이에 본 논문에서 제안하는 방법은 이와 같이 중복되는 과정을 없애고 이 메모리를 재사용하는 방법이다(그림 2).

---

1) 이는 분석한 드라이버에서의 트윅으로 일반적인 NIC 드라이버에서는 1개 소모시 1개를 다시 할당 받는 구조이다.

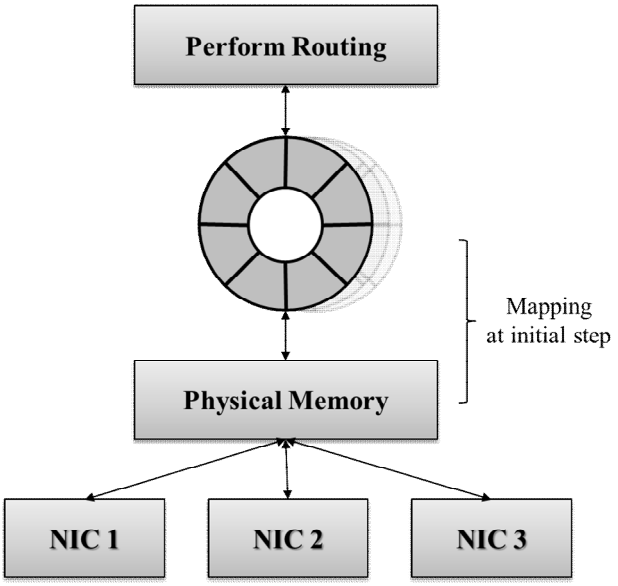

그림 2. Mapping 재사용 구조

4-1. 패킷 수신 과정

제안하는 구조는 라우터 메모리 관리를 커널에서 동작하는 라우팅 데몬에서 수행하는 구조이다. 라우팅 데몬에서 할당받은 메모리에 대해 Bounce Buffer 매핑을 미리 수행시켜서 이에 대한 핸들러를 저장시켜둔다. 이렇게 미리 저장시키는 링 버퍼를 2개 이상 할당하여 로테이션 할 수 있도록 준비한다.

할당 받은 메모리 핸들러를 NIC의 I/O Ring에 대응시켜 디바이스가 이 메모리 공간에 DMA를 수행하도록 한다. 인터럽트 처리 과정에 링 공간에 패킷을 채운다. 링이 다 채워지면 미리 할당 받은 다른 링으로 교체하고 링 전체는 메모리 접근 권한이 커널로 이관됨과 동시에 라우팅 데몬으로 전달된다.

4-2. 패킷 송신 과정

라우팅 과정을 거친 패킷은 라우팅의 결과로 다음에 전달될 인터페이스가 결정된다. 해당 인터페이스에서 접근이 가능하도록 하기 위해서는 물리 메모리에 대한 접근 권한이 주어져야 한다. 또한 DMA를 위한 물리 메모리 주소가 있어야 하는데 이미 라우팅 데몬이 초기에 소켓 버퍼 저장 공간 구성시 저장한 물리 메모리 핸들러를 알고 있으므로 이에 접근할 디바이스 권한만 새로 설정되면 이 주소를 그대로 패킷 송신에 사용될 DMA 주소로 사용하여 추가적인 패킷 복사 및 공간 할당이 필요 없어진다.

5. 결론 및 향후 연구방향

본 논문에서는 패킷 송수신 과정에서 DMA를 위한 물리 메모리 할당을 위한 Hypervisor 호출이 발생하는 성능 오버헤드에 대해서 논의하고 이를 라우팅 환경에서 최소화시키기 위한 구조를 제안하였다. 본 구조는 소프트웨어 라우터 환경에서 성능을 최대화하기 위한 구조로 라우터에서 대부분의 패킷이 수신 후 송신의 과정을 거친다는 특성을 이용한 것이다. 이와 같은 구조가 일반적인 통신 환경에 적용된다면 오히려 패킷의 메모리 복사에 대한 오버헤드가 더 커질 수 있다는 단점이 있다. 그러나 최근의 컴퓨팅 환경이 가상화된 환경에서 이루어지고 있다는 점과 이에 따른 Hypervisor 접근 오버헤드를 줄일 수 있다는 점에서 본 논문에서 제시하는 구조는 물리 메모리 관련 명령이 자주 발생할 수 있는 환경에서 또한 유효할 수 있을 것으로 보인다.

6. 참고 문헌

[1] P.Barham, I.Pratt, et al, “Xen and the Art of Virtualization”, SOSP, 2003

[2] J.W.Jang, E.Seo, H Jo, J.S.Kim, “A low-overhead networking mechanism for virtualized high-performance computing systems”, The Journal of Supercomputing, 2012

[3] YS Pan, JH Chiang, HL Li, PJ Tsao, “Hypervisor Support for Efficient Memory De-duplication”, IEEE ICPADS, 2011

[4] 이치영, 김도중, 이종원, 유혁, “라우터 가상화 환경에서의 10G 이더넷 카드를 위한 네트워크 성능 분석”, 정보과학회논문지: 정보통신 , 제 38권, 6호, pp.431-438 , Dec 2011