Xen credit 스케줄러에서의 동적 가중치 할당을 위한 성능 측정 방식 제안

이태훈 ○, 홍철호, 유혁
고려대학교
{thlee, chhong, hxy}@os.korea.ac.kr

Performance analysis for dynamic weight allocation of xen credit scheduler

Tae-hun Lee○, Cheol-Ho Hong, Chuck Yoo
Korea University

요 약

Xen의 credit 스케줄러는 서버 환경에서 도메인들의 스케줄링을 위해 설계되었다. 서버 환경의 도메인들은 네트워크 및 disk I/O가 워크로드의 대부분을 차지하지만 클라이언트 환경에서는 CPU를 포함한 다양한 워크로드의 비중이 높은 도메인들이 존재한다. 따라서 정적으로 가중치를 할당하는 경우 이러한 클라이언트 환경의 도메인들을 효과적으로 스케줄링 하기 어렵기 때문에 본 논문에서는 가중치를 동적으로 할당하는 방법을 제안하고, 보다 정확한 가중치 할당을 위한 성능 측정 방법을 연구하고자 한다.

1. 서 론

컴퓨터의 하드웨어의 성능이 발전함에 따라 시스템의 물리적인 성능도 향상되었다. 그러나 서버 시스템에서는 이 향상된 시스템 자원을 모두 사용하기 어려웠는데, 이는 서버가 필요한 시스템 자원이 일정하지 않으므로 평상시에는 시스템 자원양에 여유분을 두고 운용하기 때문이다. 따라서 서버의 성능이 증가했음에도 기존의 서버들을 통합하지 못하고 시스템 자원의 일부를 유휴 상태로 두게 되었다. 이러한 문제를 해결하기 위해 가상화 플랫폼이 등장하였는데, 이는 하나의 시스템에 두 개 이상의 운영체제를 동시에 운용하도록 하여 시스템의 자원 활용률을 높였다. 이러한 기능을 지원하는 가상화 플랫폼 중 하나인 xen은 네트워크 서버들을 대상으로 효율적인 가상화 환경을 제공한다[1].

Xen은 하나의 물리적인 하드웨어 환경을 다수의 운영체제들이 공유할 수 있도록 하는 가상 머신 모니터이다[1]. 단순히 운영체제들이 동시에 동작하는 것 뿐만 아니라, 응용 프로그램 이진 인터페이스를 유지할 수 있도록 하므로 기존의 응용 프로그램을 수정하지 않아도 실행될 수 있다. 또한 하드웨어 환경의 공유로 인해 한 운영체제의 문제가 다른 운영체제에게까지 파급되지 않도록 고립성을 지원한다.

Xen위의 가상화된 운영체제(도메인)들은 하나의 물리적인 하드웨어 자원을 공유하므로, xen이 각 도메인들에게 자원을 적절하게 분배할 수 있도록 해야 한다. 이를 위하여 xen은 도메인들을 스케줄링 하는 스케줄러를 갖고 있다. xen이 지원하는 스케줄러 중 기본적으로 사용하는 스케줄러는 Credit 스케줄러로, 도메인마다 일정한 가중치를 부여하고 부여된 가중치만큼 CPU 자원을 분배하는 방식이다. 이러한 스케줄링 정책은 도메인마다 사용 가능한 CPU 자원을 제한할 수 있으므로 공평한 자원 분배가 이루어지게 된다.

이러한 Credit 스케줄러는 서버 환경에서는 적합하지만, 클라이언트 환경에서 이를 그대로 적용하기에는 문제가 있다. 서버상에서 동작하는 도메인들의 워크로드는 대부분이 네트워크와 디스크 I/O이지만, 클라이언트 환경에서는 도메인마다 CPU를 포함한 다양한 워크로드를 갖는다. 예를 들어 계산용 프로그램은 워크로드의 대부분이 CPU에 있고, 컴파일러는 CPU 워크로드와 블록 I/O가 같이 있으며, 대용량 파일 복사 시에는 블록 I/O가 워크로드의 대부분을 차지한다. 따라서 가중치가 정적으로 고정된 기존의 Credit 스케줄러의 정책은 클라이언트 환경에 적절하지 않으므로, 이를 적합하게 수정할 필요가 있다. 이에 본 논문의 2장에서는 관련 연구, 3장에서는 기존 Credit 스케줄러를 클라이언트 환경에서 사용할 때의 문제를 밝히고, 4장에서는 동적으로 가중치를 변경하는 방식을 제안하며 5장에서는 좀 더 정확한 성능 측정 방식을 연구하고자 한다

2. 관련 연구

Xen은 반가상화 기반의 가상 머신 모니터이다[4]. 반가상화는 하부의 하드웨어와 같은 가상 머신

추상화를 제공하는 전가상화와 달리, 가상 머신 위에 올라가는 커널을 수정해야 한다. 대신 전가상화에 비해 높은 성능을 보이며, 커널 자체는 수정이 되야 하지만 응용 프로그램 이진 인터페이스의 변경은 필요하지 않으므로 응용 프로그램은 수정할 필요가 없다[1]. Xen은 domain0라는 특권 도메인을 만들어 domainU라고 부르는, xen위에서 동작하는 나머지 게스트 도메인들을 관리한다[4].

Xen의 분리된 드라이버 모델[3]은 도메인이 직접 자신의 I/O를 처리하는 것이 아니라, Isolated Driver Domain(IDD)[2] 라는 특권 도메인에서 처리한다. 게스트 도메인의 커널이 요청한 I/O는 해당 도메인의 프론트 엔드 드라이버가 받아 IDD의 백 엔드 드라이버에 다시 요청하여 IDD가 처리하도록 한다[4].

Credit 스케줄러는 도메인에게 가중치를 부여하고, 이에 따라 일정한 시간 할당량마다 도메인에 Credit를 할당한다. 할당 받은 Credit는 도메인이 VCPU를 사용하는 동안 10ms 마다 차감하고, 30ms 마다 Credit를 재할당 받는다. 도메인이 아직 Credit를 모두 소모하지 않으면 도메인의 상태는 UNDER이고, 완전히 소모하고 아직 Credit을 재할당 받지 않았다면 OVER 상태가 된다[4]. OVER상태의 도메인은 스케줄링에 제외되고 UNDER상태의 도메인들만이 CPU 자원을 할당 받아 스케줄링 된다.

3. 클라이언트 환경에서의 Credit 스케줄러

Xen의 I/O 처리방식 상 특정 도메인에서 I/O를 처리할 때 해당 도메인의 스케줄링 가중치를 소모하는 것이 아니라 IDD의 가중치를 소모하게 된다. 이렇게 도메인에 대한 I/O 작업이 IDD에서 일어나게 되면, 결과적으로 I/O를 요청한 도메인은 IDD가 대신 소모한 가중치만큼 추가적인 가중치를 얻게 되는 것과 같다. 이러한 경우 I/O 작업이 없는 도메인은 그만큼 불이익을 받게 된다. 서버상의 도메인들은 네트워크와 디스크 I/O가 워크로드의 대부분을 차지하므로 큰 영향을 받지는 않지만, 클라이언트 환경에서는 CPU 워크로드가 대부분인 도메인도 존재할 수 있는 만큼 이에 대한 고려가 필요하다.

CPU 워크로드를 갖는 도메인과 I/O 워크로드를 갖는 도메인을 비교하면, I/O 워크로드의 도메인은 I/O 시간 동안 CPU를 점유할 수 없으므로 IDD의 가중치를 소모하더라도 도메인의 가중치를 전부 소모하기 어렵기 때문에 CPU 워크로드의 도메인이 가중치에 불이익을 받지 않는다. 그러나 I/O와 CPU가 혼합된 워크로드의 도메인과 CPU 워크로드 도메인을 비교할 때, 혼합된 워크로드의 도메인은 자신의 가중치를 모두 소모하고 IDD의 가중치까지 사용할 수 있으므로, CPU 워크로드만 발생하는 도메인은 상대적으로 불평등하게 CPU 자원을 할당 받게 된다.

도메인의 워크로드에 따른 실제 CPU 사용률의 차이를 얻기 위해, 다음과 같이 실험하였다. 동일한 가중치를 부여한 네 개의 게스트 도메인 (DOM1~DOM4)을 xen 위에서 동시에 수행하면서, 하나의 도메인(DOM1)은 계산 작업을 반복하여 CPU 워크로드만을 부여하고, 다른 세 개의 도메인(DOM2~DOM4)은 동일한 계산 작업으로 CPU 워크로드를 부여함과 동시에 Iperf[6]를 사용하여 네트워크 I/O 워크로드를 부여하였다. CPU 사용률은 각 도메인이 소모한 CPU 자원양에 도메인의 I/O 요청에 의해 소모된 IDD의 CPU 자원양을 합하여 구했다. 여기서 측정된 CPU 사용률과 I/O 대역폭은 각각 [그림 1]과 [그림 2]에 나타내었다.

그림 1. CPU 사용률 (%)

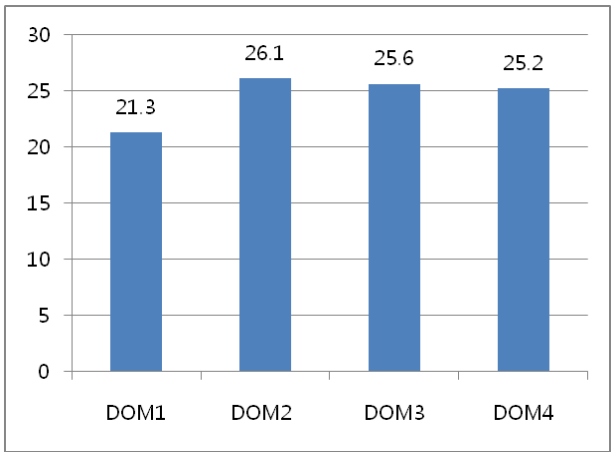
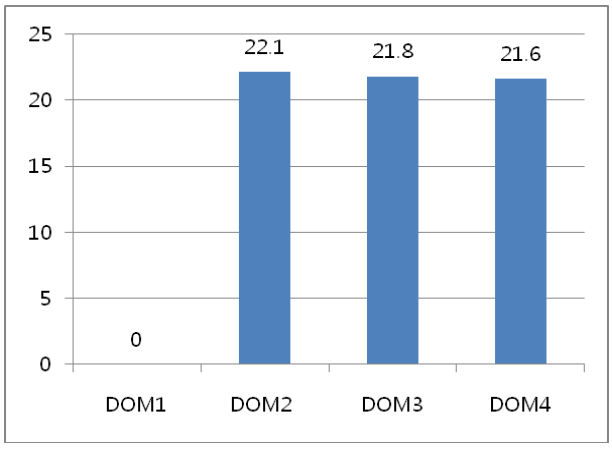


그림 2. I/O 대역폭 (Mb/s)



[그림 1]에서 볼 수 있듯이, 도메인의 가중치를 일하게 설정했음에도 불구하고 실제 CPU 사용률에는 차이가 발생했다. 이러한 가중치와 실제 CPU 사용량 사이의 차이는 가중치로 CPU 사용량을 제한하는 Credit 스케줄링 정책이 의도한 것과 다르게 발생하는 것이다. 따라서 이를 해결하기 위해, 정적인 가중치 부여 대신 I/O 요청에 의한 IDD의 CPU 소모량을 고려하여 동적인 가중치 부여 방법을 제안하였다.

4. 동적 가중치 할당 방식 제안

동적으로 가중치를 조정하기 위해, 만약 도메인이 할당된 가중치 이상으로 CPU 자원을 소비한 경우, 가중치에서 그 만큼을 차감하는 방법을 생각하였다. 도메인이 부여 받은 가중치가 도메인이 사용할 수 있는 CPU 자원의 양을 결정한다. 따라서 도메인의 I/O를 처리하기 위한 IDD의 CPU 자원도 도메인 자신의 가중치에 비례하여 분배 받아야 한다고 가정했다. 그리고 실제 I/O 사용량을 계산하여 가중치를 재분배하도록 하면 할당 받은 Credit 이상으로 사용한 CPU 자원을 반납할 수 있다.

이를 적용하여 가중치를 동적으로 할당하는 Credit 스케줄러에서의 CPU 사용률과 I/O 대역폭을 각각 [표 3]과 [표 4]에 나타내었다. 앞서 정적으로 가중치를 할당하는 기존의 Credit 스케줄러에서 측정한 것과 같이, 네 개의 도메인(DOM1~DOM4)이 xen위에서 동작할 때, 각 도메인에 동일한 가중치를 부여하고, 하나의 도메인(DOM1)은 계산작업을 반복하면서 CPU 워크로드를 부여한다. 그리고 나머지 세 개의 도메인(DOM2~DOM4) 에서는 동일한 계산작업을 반복하면서 Iperf를 사용하여 CPU와 I/O 워크로드를 부여하였다.

그림 3. CPU 사용률 (%)

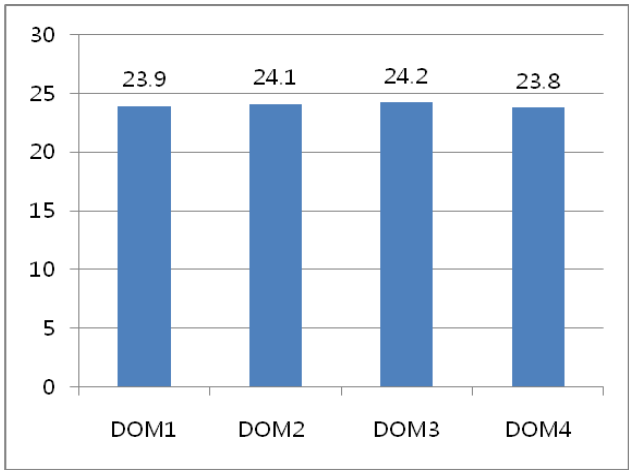
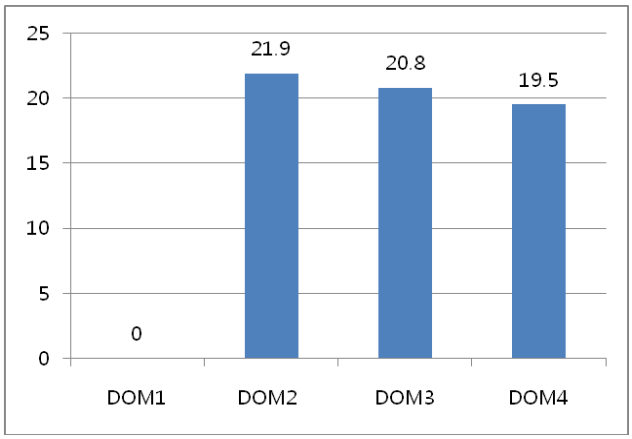


그림 4. I/O 대역폭 (Mb/s)



[그림 3]에서 볼 수 있듯, [그림 1]에 비해 CPU 사용률이 균등하게 배분되어, CPU만 소모하는 도메인이 다른 도메인에 비해 불이익을 받지 않는다. 또한 [그림 2]와 [그림 4]를 비교하면, I/O 대역폭도 약 1% 정도의 미미한 성능하락만을 보인다.

5. 성능 측정방식 제안

앞의 실험에서 도메인의 실제 CPU 사용량 측정은 I/O 발생횟수만을 고려한 단순한 방법을 사용하여 정확도가 떨어지므로 좀 더 정확한 측정방식을 제안하기로 했다.

Xen에서 구현한 “page-flipping”은 데이터 복사로 인한 오버헤드를 줄이기 위해 I/O 데이터를 포함한 IDD의 메모리 페이지를 게스트 도메인의 미사용 페이지와 교환한다[3]. 따라서 IDD와 게스트 도메인의 페이지 교환을 감지함으로써 도메인이 요청한 I/O 작업을 IDD가 수행했음을 알 수 있다[3]. 페이지 교환이 많을수록 I/O에 의해 발생한 데이터의 양도 많음을 알 수 있고, 이는 그만큼 IDD의 CPU를 많이 소모한 것으로 볼 수 있다. 따라서 IDD가 실행중인 동안 발생한 게스트 도메인과의 페이지 교환이 일어난 비율을 측정하여 게스트 도메인이 소모한 IDD의 CPU 사용량의 비율을 얻어낼 수 있다. 여기서 얻어낸 결과를 도메인의 가중치와 비교하면 도메인에 부여된 가중치 이상으로 IDD의 CPU를 초과로 소모했는지 알 수 있다. 이를 이용하여 Credit을 초기화 할 때 가중치를 어떻게 재할당 할 것인지를 결정한다.

위에서 제안한 방법은 각 게스트 도메인이 요청한 I/O의 CPU 사용량과 결과 데이터의 양 간 비율이 비슷할 경우 유용하지만, 그렇지 않은 경우에는 적합하지 않다. 따라서 I/O의 특성을 고려하여 I/O에 의한 CPU 소모량을 측정할 수 있는 방법이 필요하다. 여기서는 두 가지 I/O 워크로드인 블록 I/O와 네트워크 I/O에 대해 논하도록 하겠다.

블록 I/O는 게스트 도메인이 IDD에 요청할 때 읽거나 쓸 데이터의 양을 알려주게 된다. I/O시 발생하는 CPU 소모량은 블록 디바이스에서 읽어오거나 블록 디바이스에 쓸 데이터의 양에 비례한다. 그러므로 IDD가 일정량의 데이터를 읽고 쓸 때 소모하는 CPU 자원양을 측정하면 도메인의 블록 I/O요청에 의해 IDD가 소모하는 CPU 소비량은 도메인이 I/O를 요청한 데이터의 크기로부터 구할 수 있다.

네트워크 I/O는 블록 I/O보다 I/O와 IDD의 CPU 소모량간 상관관계를 파악하기 어려운데, 그것은 네트워크 패킷이 비동기적으로 수신측에 도달하기 때문이다[4]. 따라서 측정 방식을 간단히 하기 위해 게스트 도메인이 요청한 포트로 들어오는 패킷의 크기로부터 해당 도메인이 패킷 수신시 발생하는 네트워크 I/O의 양을 측정하도록 한다. 이를 위해서 xen이 도메인의 포트 개방 요청을 받을 때 도메인과 포트 간 관계를 저장하도록 하고, 패킷 수신시마다 참조하여 어떤 도메인에서 요청한 I/O인지를

판별하도록 한다. 패킷 송신시에도 마찬가지로 포트 번호를 통해 도메인을 판별하고, 송신하는 패킷의 크기로 네트워크 I/O의 양을 결정한다. 블록 I/O와 마찬가지로 IDD에서 네트워크 I/O를 발생시킬 때 소모하는 CPU 자원양을 측정하고, 이를 이용하여 도메인이 요청한 네트워크 I/O가 발생하는 IDD의 CPU 소모량을 계산할 수 있다.

이러한 방식은 앞서 언급한 페이지 교환횟수를 통한 측정 방식보다 좀 더 정확한 결과를 얻어낼 수 있겠지만 그 만큼 시스템 성능에 오버헤드로 작용한다. 따라서 두 방식의 정확도와 오버헤드를 비교하여 어느 쪽이 더 적합한지를 파악할 수 있도록 할 것이다.

6. 결 론

서버 환경에서 사용되었던 가상화 기술들이 클라이언트 환경에서도 확산되고 있다. 그렇지만 아직 클라이언트 환경에 최적화되지 못하고 서버 환경에 맞춰 설계된 그대로 사용하는 상태이다. Xen의 Credit 스케줄러도 서버상의 도메인들을 스케줄링 하는 데 초점이 맞춰져 있다. 이러한 이유로 클라이언트에서 동작하는 도메인들을 효율적으로 스케줄링 할 수 있도록 동적 가중치 할당방식을 제안하였다. 앞으로 본문에서 제안한 성능 측정 방식을 실제 xen 상에서 구현하여 도메인의 CPU 사용량을 좀 더 정확하게 얻어내고, 가중치를 재할당 하는 방법도 보완할 것이다. 그리고 차후 멀티코어 환경에서 동적 가중치 할당 방식을 적용할 수 있도록 확장할 계획이다.

7. 참고문헌

[1] B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, I. Pratt, A.Warfield, P. Barham, and R. Neugebauer. Xen and the Art of Virtualization. Proc. of ACM SOSP, October 2003

[2] K. Fraser, S. Hand, R. Neugebauer, I. Pratt, A. War_eld, and M. Williamson. Reconstructing I/O. Technical Report UCAM-CL-TR-596, Cambridge University, Aug 2004.

[3] L. Cherkasova and R. Gardner. Measuring CPU overhead for I/O processing in the Xen virtual machine monitor. In USENIX Annual Technical Conference, Apr 2005.

[4] H. Kim, H. Lim, J. Jeong, H. Jo, and J. Lee, “Task-aware virtual machine scheduling for i/o performance.” in *VEE ’09: Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*. New York, NY, USA: ACM, 2009, pp. 101–110.

[5] L. Cherkasova, D. Gupta, and A. Vahdat. Comparison of the Three CPU Schedulers in Xen. *ACM SIGMETRICS Performance Evaluation Review*, 35(2):42–51, 2007.

[6] Iperf: The TCP/UDP Bandwidth Measurement Tool. http://dast.nlanr.net/Projects/Iperf

[7] D. Chisnall. *The Definitive Guide to the Xen Hypervisor*. Prentice