

연성 실시간 Xen 하이퍼바이저를 위한 실시간 스케줄러 설계 및 구현

허경우, 김병기, 장재혁, 고영웅
한림대학교 컴퓨터공학과

e-mail:{rapperkw, bkkim, jaehyok2, yuko}@hallym.ac.kr

Design and Implementation of Real-time Scheduler for Soft Real-time Xen Hypervisor

Kyung Woo Hur, Byung Ki Kim, Jae Hyeok Jang, Young Woong Ko
Dept of Computer Engineering, Hallym University

요 약

본 논문에서는 Xen 하이퍼바이저에서 연성 실시간 작업을 지원하기 위한 실시간 스케줄링 알고리즘을 제안하고 있다. 제안하는 기법은 하이퍼바이저에서 제공하는 Credit 스케줄러를 개선하여 실시간 게스트 운영체제의 VCPU를 우선적으로 처리하도록 하였다. 또한 실시간 VCPU가 실시간 처리를 위해 요구하는 프로세서 용량을 제외한 부분 내에서 비실시간 VCPU가 수행 될 수 있도록 프로세서 용량을 제한하였다. 실험을 통하여 제안하는 실시간 스케줄러가 실시간 태스크의 데드라인 실패율을 크게 줄임을 알 수 있다.

1. 서론

최근 가상화(Virtualization) 기술은 클라우드 컴퓨팅과 그린 IT를 위한 기반 기술로 연구 및 산업 분야에서 많은 관심을 받고 있다. 가상화 기술은 하나의 물리적인 시스템에서 자원을 동적으로 공유 및 재구성 하는 기술을 제공한다. 그래서 시스템 자원의 활용률을 높일 수 있고 이를 통해 시스템 유지비용을 낮추는 효과가 있다. 최근에는 가상화 환경에서 임베디드 소프트웨어 및 스트리밍 서버와 같이 시간 제약적인 작업 처리를 수행하려는 요구가 증가 하고 있다. 하지만 가상화 기술은 내재적인 여러 가지 이유로 인해 실시간(Real-time) 작업처리에 어려움이 많으며 이를 해결하기 위해 다양한 연구가 진행 되고 있다.[1,2,3,4]

본 논문에서는 오픈소스 가상화 프로젝트인 Xen[5] 환경에서 게스트 운영체제의 실시간 태스크 처리를 지원하기 위해 실시간 스케줄러를 설계하고 구현하였다. 제안하는 방법은 두 가지 특징을 가진다. 먼저 실시간 태스크를 수행하는 게스트 운영체제의 VCPU(Virtual CPU)를 우선적으로 스케줄링 한다. 또한 실시간 게스트 운영체제의 VCPU 수행시간을 보장하기 위하여 비실시간 게스트 운영체제의 VCPU 프로세서 용량을 제한한다. 본 연구에서는 실험을 통해 제안하는 방식이 비실시간 게스트 운영체제의 태스크 처리량을 크게 감소시키지 않고 실시간 작업을

효과적으로 지원함을 보였다.

2. 스케줄러 설계 및 구현

본 연구에서 제안하는 가상화 환경은 오픈소스로 공개되어있는 Xen을 사용하였다. Xen은 2003년 영국 캠브리지 대학에서 개발한 가상화 프로젝트로 사용 및 분석이 용이하다. Xen은 VMM(Virtual Machine Monitor) 혹은 하이퍼바이저(Hypervisor)로 불리는 가상의 레이어에서 게스트 운영체제가 동작한다. 하이퍼바이저는 게스트 운영체제에게 가상의 CPU인 VCPU를 할당해주고 스케줄링 정책에 따라서 VCPU를 실제 물리 CPU에 맵핑하여 자원을 제공한다. Xen에는 기본적으로 Credit 스케줄러가 설정되어 있으며 Credit 스케줄러는 *credit* 이라는 특정 값을 이용하여 VCPU의 우선순위를 결정한다.

본 논문에서는 Xen 시스템의 Credit 스케줄러를 수정하여 실시간 게스트 운영체제의 응답성을 높이고 CPU 자원을 보장하였다. 그림 1은 제안하는 실시간 스케줄러의 동작 원리를 보이고 있다. 기존의 Credit 스케줄러는 실시간 처리의 여부에 관계없이 공평하게 CPU 자원을 분배하기 위해 설계 되어있다. VCPU의 우선순위가 같다면 VCPU가 CPU자원을 모두 소비하고 다시 런큐에 삽입될 때 동일한 우선순위를 가진 VCPU 중 제일 마지막 위치에 삽입된다. 그림에서 Credit RunQ를 보면 VCPU1이 실행을 마치고 다시 런큐에 삽입될 때 VCPU3 뒤에 위치하는 것을 알 수 있다. 이 방법은 VCPU 사이의 공평성은 보장해

이 논문은 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No.2010-0005442)

줄 수 있지만 실시간 VCPU의 응답성이 떨어진다. 본 연구에서 제안하는 스케줄러는 실시간 여부를 판단하여 수행 중인 VCPU가 실시간 VCPU이면 같은 우선순위 중에 실시간 VCPU 중의 가장 마지막, 비실시간 VCPU 중에 가장 앞부분에 위치하게 된다. RT-Credit RunQ를 보면 기존의 Credit 스케줄러의 RunQ와는 다르게 VCPU2의 뒤에 다시 삽입되는 것을 볼 수가 있다.

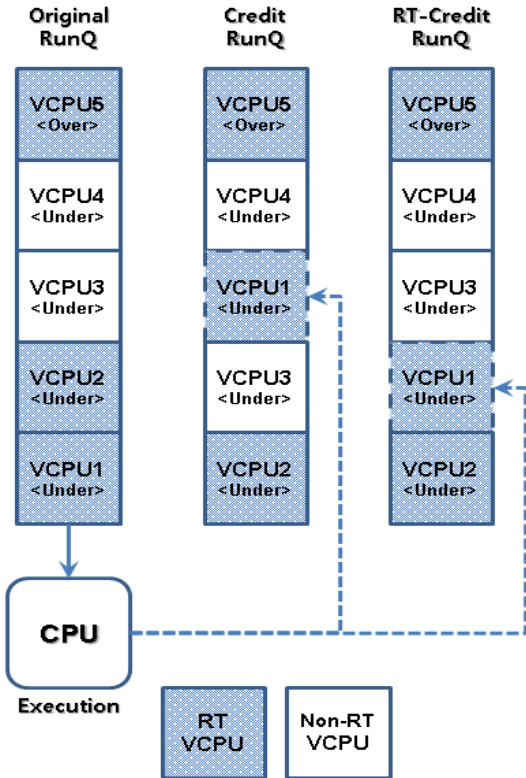


그림 1. 제안하는 RT-Credit 스케줄러 개념도

본 연구에서는 추가적으로 실시간 VCPU의 CPU 자원을 보장하기 위해서 비실시간 VCPU의 프로세서 용량을 모니터링 하고 제한한다. 하나의 CPU 코어에서 실시간 게스트 운영체제와 비실시간 게스트 운영체제가 동시에 수행하고 있을 때 프로세서 용량의 합이 100%를 초과하면 안된다. 또한 실시간 VCPU가 사용해야 할 프로세서 용량을 제외한 나머지 부분을 비실시간 VCPU가 사용해야 실시간 VCPU의 실시간 처리를 보장할 수 있다. 그래서 본 연구에서는 기존의 Credit 스케줄러에서 CPU 자원을 제한 할 수 있는 *cap* 값을 동적으로 변경하는 모듈을 구현하였다. 비실시간 VCPU의 *cap* 값을 현재 CPU에서 수행되고 있는 실시간 VCPU의 상황에 맞게 제한하여 실시간 VCPU의 실시간 처리를 보장하였다.

3. 실험 결과 및 분석

제안하는 시스템의 성능 측정을 위해 Intel Core i7 920 2.66GHz, DDR3 6GB, Seagate 1TB 7200RPM 인 하드웨어 플랫폼에서 실험을 수행하였다. 실험의 편의를 위해

CPU Core 는 2개를 사용하였으며 하이퍼스레딩 기능은 사용하지 않았다. 소프트웨어는 Xen 3.4.3, CentOS 5 커널 버전 2.6.18.8을 사용하였으며 성능을 측정하기 위해 두 가지의 워크로드를 생성하였다.

- 실시간 워크로드, $T(Period, Execution\ time)$: 실시간 운영체제의 성능을 평가하기 위해서 생성된 워크로드이다. 주기(Period)와 수행시간(Execution time)을 가지고 동작한다. 수행시간은 MD5 해시 함수를 이용하여 메모리에 있는 일정한 크기의 데이터를 해싱한다. 한 번의 해싱에 1ms 시간이 걸리도록 메모리의 크기를 조정하고 수행시간 만큼 반복하여 수행하게 한다. 주어진 주기 내에 작업이 수행되지 못하면 데드라인을 놓친 것으로 판단하여 수행하지 못한 횟수를 로그로 저장한다.
- 비실시간 워크로드, $T(Execution\ time)$: 비실시간 운영체제의 성능을 평가하기 위해 생성한 워크로드로서 일정한 크기의 메모리 블록을 MD5 해시 함수로 무한 반복한다. 그리고 실험이 진행되는 동안 해시함수를 수행한 횟수를 로그로 저장한다. 이를 통해 비실시간 워크로드의 처리량을 계산 할 수 있다.

본 실험에서는 Xen에서 입출력과 게스트 운영체제를 관리하는 Dom-0의 영향을 줄이기 위해 0번 코어에 고정(pinning)시키고 실험을 진행하였다. 실시간 게스트 운영체제들은 $T(100,70)$ 의 워크로드를 수행하고, 비실시간 게스트 운영체제에는 $T(60)$ 의 비실시간 워크로드를 수행하였다. 스케줄러의 성능 비교를 위해 사용한 스케줄링 정책은 다음과 같다.

- Default Credit : Xen 하이퍼바이저의 기본 스케줄러로 모든 설정 값은 기본상태로 실험하였다.
- RT Pinning : Credit 스케줄러에서 실시간 게스트 운영체제를 CPU 코어에 고정시켰다. Credit 스케줄러는 로드 밸런싱 기능이 있어서 CPU 코어에 고정된 Dom-0와 실시간 게스트 운영체제를 제외한 나머지 게스트 운영체제만 마이그레이션 된다.
- RT-Credit : 본 연구에서 제안하는 실시간 스케줄러이며 실시간 게스트 운영체제의 응답성 및 비실시간 게스트 운영체제의 VCPU 처리 용량을 모니터링 하고 제한하는 기능을 가진다.

그림 2는 각 스케줄러 별로 실시간 VCPU가 실시간 워크로드를 수행한 결과를 나타내고 있다. X축은 데드라인 실패 횟수를 나타내고 있으며 Y축은 태스크의 개수를 의미한다. X축이 0인 경우에는 데드라인 실패가 없음을 의미하며 RT-Credit 의 대부분의 태스크가 데드라인의 실패 없이 수행이 완료 되었음을 보인다. 나머지 스케줄러 들은 약 20개 내외만 성공을 하고 나머지는 모두 데드라인을 놓친 것을 볼 수 있다. Credit 스케줄러와 RT-Pinning 스

케줄러의 정책은 거의 동일 한 결과를 보여주고 있으며 본 연구에서 제안한 RT-Credit 스케줄러가 다른 정책들 보다 좋은 실시간 처리를 성능을 보이고 있다.

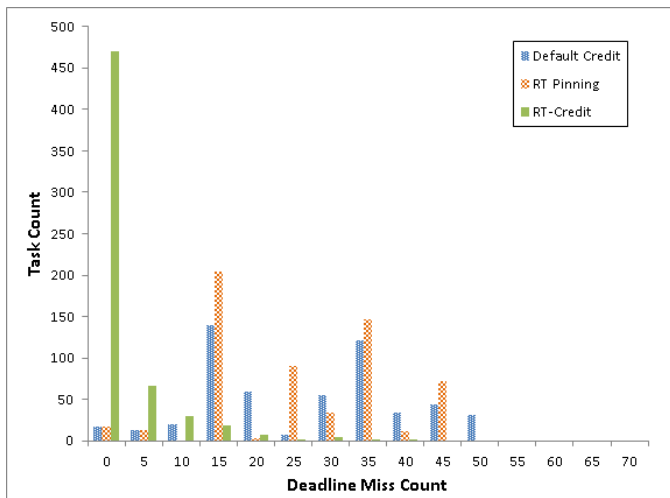


그림 2. 데드라인 실패율 그래프

그림 3은 각 스케줄러 별로 실험을 수행하였을 때 비실시간 워크로드의 처리량을 나타내고 있다. RT-Credit 스케줄러 방식이 다른 스케줄러 방식에 비해서 처리량이 다소 감소 함을 볼 수 있는데 이는 실시간 VCPU의 실시간 처리를 위해서 비실시간 VCPU의 프로세서 용량을 제한하기 때문이다. Credit 스케줄러와 RT-Pinning 스케줄러는 높은 처리량을 보였지만 상대적으로 그림 2에서는 실시간 태스크의 데드라인 실패율이 많은 것을 볼 수 있다.

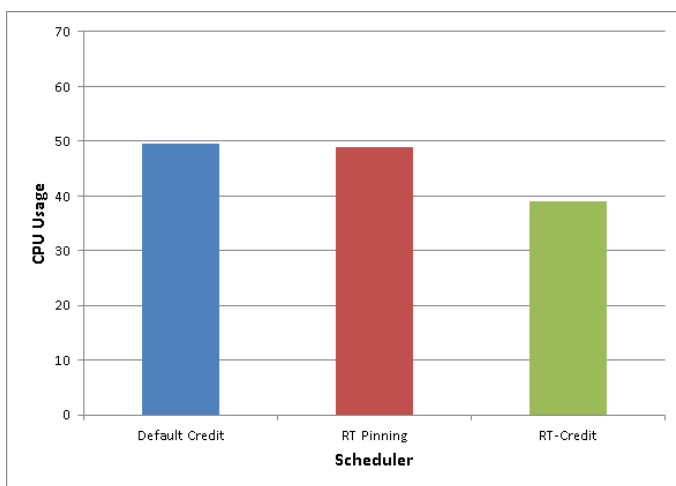


그림 3. 비실시간 워크로드에 대한 처리량 실험 결과

4. 결론 및 향후 연구

본 논문에서는 Xen 가상 머신 플랫폼에서 실시간 게스트 운영체제의 실시간 처리를 지원하기 위하여 실시간 게스트 운영체제에게 할당된 VCPU의 우선순위를 높이고,

비실시간 VCPU의 프로세서 용량을 모니터링하고 제한하는 방식을 적용하여 연성 실시간 처리를 지원하는 방식을 제안하였다. 제안하는 방식의 아이디어 및 구현 방식을 기술하였고 실험을 통해 유용성을 보였다. 실험 결과를 통해 실시간 게스트 운영체제의 실시간 태스크가 대부분 데드라인을 놓치지 않고 처리되는 것을 보였다. 하지만 실시간 처리를 위해 비실시간 게스트 운영체제의 프로세서 용량을 제한하는 기법이 비실시간 태스크의 처리량을 다소 감소 시키는 문제를 발생시켰으며 향후 이와 같은 처리량 감소를 효율적으로 개선하는 연구를 수행할 계획이다.

참고문헌

[1] Huacai Chen, Hai Jin, Kan Hu, Minhao Yuan, "Adaptive Audio-aware Scheduling in Xen Virtual Environment", Computer Systems and Applications (AICCSA), 2010 IEEE/ACS International Conference, Hammamet, 2010

[2] Min Lee, A. S. Krishnakumar, P. Krishnan, Navjot Singh, and Shalini Yajnik. "Supporting soft real-time tasks in the xen hypervisor." In VEE '10: Proceedings of the 6th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, pages 97 - 108, New York, NY, USA, 2010. ACM.

[3] Robert Kaiser. "Alternatives for scheduling virtual machines in real-time embedded systems." In IIES '08: Proceedings of the 1st workshop on Isolation and integration in embedded systems, pages 5 - 10, New York, NY, USA, 2008. ACM.

[4] Gernot Heiser. "The role of virtualization in embedded systems." In IIES '08: Proceedings of the 1st workshop on Isolation and integration in embedded systems, pages 11 - 16, New York, NY, USA, 2008. ACM.

[5] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. "Xen and the art of virtualization." In SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles, pages 164 - 177, New York, NY, USA, 2003.ACM.