

# 게스트 운영체제 실시간성 보장 측면에서의 Xen 분석

(An Analysis on Xen in View of Real-time Guarantee)

박미리\*, 유시환, 유혁  
(Miri Park, See-hwan Yoo, Chuck Yoo)

Abstract : Embedded systems have been a target of virtualization for recent years. In this paper we analyze Xen virtualization technology in view of real-time guarantee. Firstly, we show the credit scheduler is not suitable to allocate CPU for real-time guest OS because it does not consider the timing constraints. SEDF scheduler with compositional framework would provide the guaranteed service. Secondly, we argue the split device driver model makes the system unpredictable. Asynchronous inter-domain communication and unexpected processing time of I/O model do not give the determinism. Thus native driver of real-time guest OS should be allowed to process I/O without virtualization. It is possible under the assumption that the certain devices are dedicated to real-time VM.

Keywords : real-time, virtualization, embedded

## 1. 서론

서버 분야에서 널리 이용되고 있는 가상화(virtualization) 기술이 최근 들어 임베디드 시스템에도 적용되려 하고 있다. 한두 가지 특화된 기능만을 하던 종래의 디지털 기기들은 이제 다양한 기능을 함께 탑재하려는 디지털 컨버전스 요구에 응하고 있으며, 발전하는 임베디드 시스템의 성능 또한 이런 환경을 기술적으로 뒷받침하고 있다. 그러나 대부분의 실시간 운영체제(real-time operating system)의 경우 특정 환경에 제한적으로 설계되어 있으므로 시장의 다양한 요구 사항을 충분히 반영하기에는 부족하다. 가상화 기술은 실시간 운영체제와 더불어 범용 운영체제 등의 다른 운영체제를 하나의 물리 머신 위에서 동작하도록 지원하여 이러한 제약 사항을 뛰어넘게 한다.

보안상의 목적 또한 가상화 기술을 임베디드 시스템에 도입하려는 이유 중 하나이다.[1] 많은 디지털 기기들이 사람들의 일상생활에서 이용됨에 따라 이들에 대한 보안의 중요성이 부각되고 있다. 각 VM(Virtual Machine)에 고립성(isolation)을 제공하는 것은 가상화 개념이 제시하는 기본 요건 중 하

나이며, 이는 보안 문제를 해결하기 위해 이용될 수 있다.

임베디드 시스템에 이용되고 있는 arm 프로세서를 가상화의 대상으로 삼으려는 연구는 [2]과 [3] 등을 통하여 진행되어 왔으며 최근 들어서는 [4]와 [5]에서 임베디드 시스템을 위한 가상 머신 모니터(Virtual Machine Monitor, VMM)를 제시하였다. 특히 [6]에서는 서버 가상화 분야에서 널리 쓰이고 있는 Xen을 arm 아키텍처에 적용하였다.

임베디드 시스템은 일반적으로 실시간 운영체제가 자원을 관리한다. 따라서 기존의 가상화 연구에서는 고려되지 않았던 보장된 응답성이 새로운 문제점으로 떠올랐다. 기존의 VMM이 각 VM에 공평(fair)한 자원을 주는 것을 중요한 스케줄링 목표로 삼는다면, 임베디드 시스템의 VMM은 게스트 운영체제가 실시간성을 보장할 수 있도록 시간 제약을 고려한 자원 분배를 해야 한다. 실시간 운영체제는 각 태스크가 특정 시간 내에 수행을 마치도록 보장하는 시간 조건(timing constraints)을 중요한 요구 사항으로 여기며 주어진 자원을 이에 맞게 예측하고 관리 할 수 있어야 한다. 이와 같은 특성은 VMM이 제공하는 가상 머신 위에서도 그대로 지켜져야 할 것이다.

본 논문에서는 반가상화(para-virtualization)을 통해 성능 향상에 기여한 Xen[7]을 게스트 운영체제의 실시간성 보장 측면에서 분석하고자 한다.

\* 교신저자(Corresponding Author)

박미리, 유시환, 유혁 : 고려대학교 컴퓨터학과

이어지는 2장에서는 우리가 가정하고 있는 시스템에 대한 설명과 연구 범위에 대해 소개한다.

## II. 시스템 제약 사항과 연구 범위

우리는 이번 연구에서 두 개의 게스트 운영체제를 가정하였다. 두 개의 VM에 각각 실시간 운영체제와 범용 운영체제가 동작하는 것으로 간주한다. 이러한 가정은 모바일 기기에서 디지털 시그널을 처리하는 코어 부분을 위한 실시간 운영체제와 이에 더불어 추가되는 다양한 기능을 위한 범용 운영체제를 함께 동작시키는 경우에 해당된다.

이 시스템에 올라가는 실시간 운영체제는 실시간 커널이 가지는 요건을 갖추어, 물리 머신에서 동작시켰을 때 실시간성을 보장할 수 있다고 가정한다. 우리의 관심은 기존에 실시간성을 보장하며 동작하던 실시간 운영체제가 Xen 위에서 범용 운영체제와 함께 동작하면서도 실시간성이 보장될 수 있는가 하는 것이다.

[8]은 실시간 태스크를  $t(p,e)$ 로 정의하였는데 이는 태스크를 주기성을 가지고 실행 요구되어 특정 시간 내에 실행을 마치는 것으로 본다. 즉, 주기  $p$ 마다 릴리즈 되는 태스크  $t$ 가 있다면 릴리즈 될 때마다 상수 시간  $e$ 만큼 실행이 되어야 한다. 이때 다음 릴리즈 될 때까지 실행이 완료되어야 하므로 데드라인은  $p$ 가 된다.

이러한 정의를 바탕으로 모든 태스크의 실시간성을 보장하기 위해서는 각 태스크에 적절한 우선순위를 주어 스케줄링 함으로써 데드라인을 놓치지 않도록 하는 것이 중요하다. 대표적인 실시간 스케줄링 방식인 RM(Rate Monotonic)과 EDF(Earliest Deadline First) 스케줄러는 실시간 태스크에 적절한 우선순위를 주어 데드라인을 놓치지 않도록 한다.

실시간 운영체제에서는 I/O 처리 모델 역시 고려해야 한다. I/O 자체도 실시간성을 가지게 하기 위하여 응답 시간을 짧게 함과 동시에, 처리 모델의 실행이 예측 가능하게 하기 위해 결정성을 제공해야 하기 때문이다.

다음에 나오는 3장과 4장에서는 각각 Xen의 스케줄러와 I/O처리 구조에 대해 분석하여 Xen이 게스트 운영체제의 실시간성을 보장하기 위한 방안에 대해 논의한다.

## III. XEN 스케줄러

Xen은 3.0버전 현재 대표적인 스케줄러로 크레

딧스케줄러(credit scheduler)와 SEDF 스케줄러(Simple Earliest Deadline First scheduler)를 지원하고 있다. 기본 스케줄러인 크레딧 스케줄러는 각 VM에 일종의 가중치인 크레딧을 주고 크레딧이 정의한 만큼 자원을 할당한다. 크레딧은 시간 제약을 고려하지 않은 것으로 CPU자원을 얼마나 나누어 주어야 하는 지에 대해서만 고려하고 있을 뿐, 언제 주어야 하는지는 고려하고 있지 않다. 따라서 이와 같은 방식으로 VM을 스케줄링 할 경우에 각 VM의 실시간 태스크들은 실시간성을 보장받기 어렵다. 만약 실시간 운영체제가 자신이 관리하고 있는 모든 태스크의 실시간성을 고려하고 있다고 해도 다른 VM에 수행이 넘어가 자신이 CPU자원을 할당 할 수 없는 상황이라면 데드라인이 다가오는 태스크는 수행을 하지 못하게 된다. 따라서 실시간 운영체제가 요구하는 시간 제약성은 지켜 지지 않는다.

[9]는 VM을 각각 time-driven real-time, event-driven real-time, non real-time 작업만을 수행하는 것으로 제한 시켜 실시간 VM에 높은 우선순위를 부여하여 스케줄링 하는 기법을 제시하였다. 이 기법을 적용할 경우 위와 같은 극히 제한된 환경에서 실시간 스케줄링이 가능하다. 그러나 우선순위를 실시간 VM에 높게 준다고만 밝혔으므로 실시간성을 보장 할 수 있는 메커니즘은 제공되지 않는다. 특히나 특정 VM에 무조건적으로 높은 우선순위를 부여하는 것은 자칫 다른 VM의 기아(starvation) 현상을 초래할 수도 있다.

Xen은 또 하나의 다른 스케줄러인 SEDF 스케줄러를 제공한다. 이는 [8]에서 제시한 것처럼 주기당 실행 시간으로 정의된 태스크 중에서 데드라인에 가장 가까운 것에 높은 우선순위를 주어 스케줄링한다. 따라서 게스트 운영체제들이 필요로 하는 CPU자원을 주기성을 가진 자원  $g(p,e)$ 로 제시한다면 VM 레벨의 실시간성을 보장할 수 있다.

[10]이 보여준 복합적인 프레임워크는 서로 다른 실시간 스케줄링 방식을 쓰는 컴포넌트가 각각 필요로 하는 자원을 모델링한다. 이를 적용한 계층적인 스케줄링 방식은 그림1과 같이 다양한 게스트 운영체제의 CPU 자원을 모델링 할 수 있게 한다. 게스트 운영체제 G1은 자신이 관리하고 있는 모든 실시간 태스크들이 필요로 하는 자원을 주기성을 가진 자원  $G1(g1_p, g1_e)$ 으로 하이퍼바이저에게 요구한다. 하이퍼바이저는 이에 따라 VCPU를 할당해주고, 수행을 넘겨받은 게스트 운영체제는 자신의 실시간 스케줄러를 이용하여 다시 태스크를 스케줄링 한다.

따라서 각 게스트 운영체제가 필요로 하는

CPU자원은 하이퍼바이저의 스케줄링에 의하여 제공되고 실시간 태스크의 실행은 해당 게스트 운영체제가 실시간 스케줄링을 통하여 보장할 수 있다.

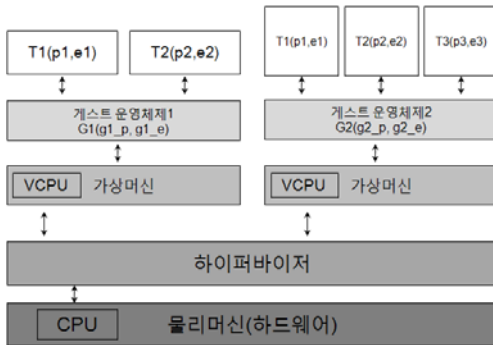


그림 1. 계층적 스케줄링

#### IV. XEN I/O 처리 구조

실시간 시스템이 갖추어야 할 요건 중 하나는 실행이 결정적(deterministic)이어야 한다는 것이다. 결정적이라는 것은 시스템의 실행이 실행 전에 미리 예측 가능해야 함을 의미한다. 스케줄러가 데드라인을 고려하여 태스크를 스케줄링 한다고 해도 커널을 비롯한 소프트웨어 전체의 실행을 컨트롤하지 못한다거나, 예측할 수 없다면 전체적인 실시간 보장은 어렵게 된다. 특히 인터럽트의 경우 대체로 높은 우선순위를 가지기 때문에 이들이 실행 중인 실시간 태스크를 선점하여 수행됨은 자칫 예상했던 태스크의 실행을 보장하지 못하는 결과를 초래하기도 한다.

Xen은 그림2와 같이 디바이스 드라이버를 도메인으로 분리하는 분리 드라이버 모델(split driver model)을 채택하고 있다. 디바이스 드라이버를 분리하는 것은 디바이스 드라이버의 문제점을 고립화(fault isolation)시키고 동시에 관리 유지하기 쉽게 하는 장점을 지니고 있다. 이는 많은 문제점을 야기한다고 알려진 드라이버의 이슈를 전체 시스템의 실패로 이어지지 않게 한다.

하지만 이와 같은 구조는 I/O 처리에 대한 예측을 어렵게 만드는 요인이 된다. 일반적으로 실시간성을 고려하지 않는 범용 운영체제는 I/O처리를 하는 것에 대한 예측이나 보장을 제공하지 않는다. 따라서 이런 범용 운영체제의 드라이버가 분리 드라이버 도메인으로 사용 된다면 전체 시스템의 I/O 처리에 대한 결정성은 보장되기 어렵다. 이는 실시간성을 지원해야 하는 I/O에 대한 컨트롤이 힘들게 하는 요인이 된다.

하지만 특정 디바이스를 실시간 운영체제에서 독점적으로 사용한다고 가정한다면 이 문제는 I/O 처리 모델을 수정함으로써 개선될 수 있다.

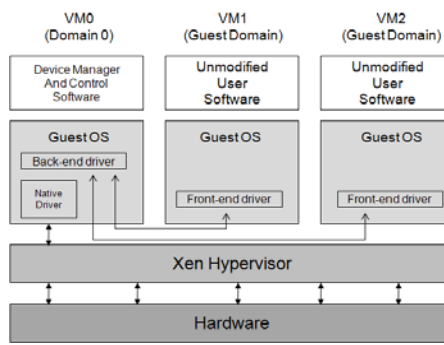


그림 2. Xen의 분리된 드라이버 도메인 구조([11] 인용)

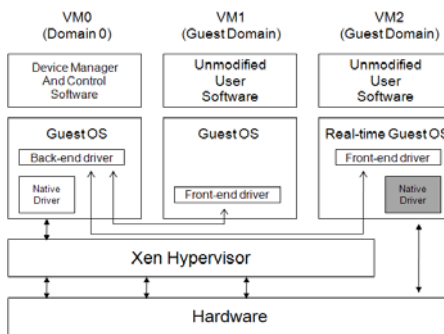


그림 3. 가상화를 거치지 않는 I/O 처리가 추가된 모델

그림 3과 같이 특정 디바이스가 실시간 운영체제에 의해 독점적으로 사용된다면, 이 자원은 가상화를 거치지 않고 직접 처리될 수 있게 할 수 있다. [12]에서 제시한 방식은 드라이버 도메인을 거치는 기존의 I/O 처리 모델을 게스트 운영체제의 디바이스 드라이버가 직접 처리하게 한다. 이와 같이 가상화를 거치지 않는 I/O 처리 기법은 성능상의 이득을 얻게 한다. 또한 실시간 운영체제가 가지고 있는 디바이스 드라이버는 기존의 예측성과 결정성을 그대로 유지하고 있으므로 I/O처리와 실시간 운영체제 자체의 실행도 보장해 줄 수 있을 것이다.

#### V. 결론 및 향후 계획

Xen의 스케줄러와 I/O 처리 모델은 게스트 운영체제의 실시간성을 보장하는 데에 가장 중요한 요

소이다. 스케줄러는 각 VM의 게스트 운영체제가 실시간성을 보장할 수 있도록 필요한 자원을 할당해 주어야 한다. 이때 시간 제약성을 고려한 실시간 스케줄러가 지원되어 실시간성을 보장함과 동시에 다른 게스트 운영체제가 기아 상태에 빠지지 않도록 컨트롤 할 수 있어야 한다. I/O처리는 우선 I/O 처리 자체의 응답시간을 높임과 동시에 실행이 예측 가능하도록 하여야 한다. 하지만 현재 구조로는 비동기적인 통신 방식과 분리된 도메인 드라이버에 따르는 구조상 예측이 불가능하다. 따라서 실시간 운영체제 자체에서 I/O처리를 하도록 해야 한다. 이런 접근은 우리가 구상한 시스템에서 실시간 운영체제가 독립적으로 사용하는 디바이스에 대한 가정을 함으로써 가능하다.

본 논문에서는 게스트 운영체제의 실시간성 보장을 위해 Xen 구조를 분석하였다. 앞으로의 연구에서는 이를 바탕으로 한 실험을 통해 실시간 운영체제의 성능을 정량적으로 보여야 할 것이다.

### 참 고 문 헌

- [1] J. Brakensiek, A. Droge, H. Hartig, A. Lackorzynski, and M. Botteck, "Virtualization as an enabler for security in mobile devices," in Workshop on Isolation and Integration in Embedded Systems, IIES, 2008. Glasgow, UK. 1st ACM SIGOPS, April 2008.
- [2] D. R. Ferstay, "Fast secure virtualization for the arm platform," Master's thesis, University of British Columbia, 2006.
- [3] R. Bhardwaj, P. Reames, R. Greenspan, V. S. Nori, and E. Ucan, "A Choices Hypervisor on the ARM architecture," University of Illinois, Urbana-Champaign, 2006, cS523 Course Project Report.
- [4] Trango: secured virtualization on ARM, Trango, <http://www.trango-vp.com>.
- [5] VirtualLogix Real-Time Virtualization and VLX, VirtualLogix, <http://www.osware.com>.
- [6] J.-Y. Hwang, S.-B. Suh, S.-K. Heo, C.-J. Park, J.-M. Ryu, S.-Y. Park, and C.-R. Kim, "Xen on arm: System virtualization using xen hypervisor for arm-based secure mobile phones," Consumer Communications and Networking Conference, 2008.
- [7] Paul Barham , Boris Dragovic , Keir Fraser ,

Steven Hand , Tim Harris , Alex Ho , Rolf Neugebauer , Ian Pratt , Andrew arfield, "Xen and the art of virtualization," Proceedings of the nineteenth ACM symposium on Operating systems principles, October 19-22, 2003.

- [8] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," J. ACM, vol. 20, no. 1, pp. 46 - 61, 1973.
- [9] R. Kaiser, "Alternatives for Scheduling Virtual Machines in Real-Time Embedded Systems" 1st Workshop on Isolation and Integration in Embedded Systems, April 2008
- [10] I. Shin , I. Lee, "Compositional Real-Time Scheduling Framework," Proceedings of the 25th IEEE International Real-Time Systems Symposium, p.57-67, December 05-08, 2004
- [11] I. Pratt. Xen Virtualization. Linux World 2005 Virtualization BOF presentation
- [12] Jiuxing Liu , Wei Huang , Bulent Abali , Dhabaleswar K. Panda, "High performance VMM-bypass I/O in virtual machines," Proceedings of the Annual Technical Conference on USENIX'06 Annual Technical Conference, p.3-3, May 30-June 03, 2006

### 저 자 소 개

**박 미 리**(Miri Park)

2006년 : 고려대학교 컴퓨터학과 학사

2008년~현재 : 고려대학교 컴퓨터학과 석사 과정

관심분야 : 실시간 운영체제, 가상화 기술

Email : mrpark@os.korea.ac.kr

**유 시 환**(See-hwan Yoo)

2002년 : 고려대학교 컴퓨터학과 학사

2004년 : 고려대학교 컴퓨터학과 석사

2004년~현재 : 고려대학교 컴퓨터학과 박사 과정

관심분야 : 임베디드 소프트웨어, 실시간 운영체제

Email : shyoo@os.korea.ac.kr

**유 혁**(Chuck Yoo)

1982년 : 서울대학교 학사

1986년 : University of Michigan 컴퓨터 공학 석사

1990년 : University of Michigan 컴퓨터 공학 박사

1995년~현재 : 고려대학교 컴퓨터학과 교수