

# 가상화 기법을 이용한 리눅스 기반 임베디드 시스템의 성능 분석 및 검증 도구

곽상헌\*, 임성수\*\*

국민대학교 컴퓨터공학과

e-mail : hunnyk7@kookmin.ac.kr\*, sslim@kookmin.ac.kr\*\*

## Performance analysis and testing tool for linux based embedded system with virtualization techniques

Sangheon Kwak\*, Sung-Soo Lim\*\*

Dept. of Computer Science, Kook-min University

### 요 약

본 논문에서는 성능 분석과 검증을 위한 방법으로 가상화 기법을 이용하여 성능을 분석할 수 있는 도구를 제안한다. 가상 머신의 성능 분석을 통해 원하는 이벤트들의 발생 양상에 따른 시스템의 성능을 호스트 머신에서 파악할 수 있다. 즉 가상 머신에서 사용하는 자원과 발생하는 이벤트에 대한 정보를 호스트 머신에서 확인할 수 있고, 가상 머신에서 발생한 사용자 입력을 호스트 머신이 임의로 재생시킬 수 있다. 이러한 기능을 통해 사용자 입력 패턴에 따른 시스템 자원의 상태 및 성능을 분석하여, 해당 시스템의 안정성을 시험할 수 있는 검증 환경을 제공한다.

### 1. 서론

임베디드 시스템의 하드웨어와 소프트웨어가 복잡해지면서, 시스템의 안정적인 운영에 대한 요구가 늘어나고 있다. 즉, 제한된 자원을 효율적으로 이용하는지의 여부를 성능 분석을 통해 파악하는 일이 더욱 중요해지고 있는 것이다.

가상화(Virtualization)란 하나의 호스트 머신(Host Machine, HM)상에서 복수개의 가상 머신(Virtual Machine, VM)이 동작하게 하는 기법을 말한다. 가상 머신은 자신이 모든 하드웨어를 독점하여 운영되는 것처럼 동작한다. 이것은 복수개의 운영체제 탑재나 보안의 목적으로 임베디드 시스템에서 많이 적용되고 있는 기술이다. 이러한 기법을 임베디드 시스템을 가상으로 검증하기 위한 방법으로 사용하는 연구가 이루어지고 있다([1], [2]). 즉, 실제 하드웨어를 가상화하여 동작시키고, 가상 머신의 성능을 분석하여 검증하는데 이용하는 것이다.

본 논문에서는 가상화 기법을 이용하여 리눅스 기반의 임베디드 시스템의 성능을 분석하고 검증할 수 있는 도구를 제안한다. 제안하는 도구는 시스템 소프트웨어를 검증하기 위해 반복되는 컴파일, 빌드, 다운로드, 실행과 같은 일들을 호스트 머신이 가상화하는 가상 머신에서 가능하게 하여, 편리해진 검증 환경을 제공한다. 또한 이벤트를 관리하는 기능을 제공하여 특정 시스템 소프트웨어의 이벤트에 대한 성능을 검증할 수 있도록 하였다.

실제 시스템과 유사한 하드웨어 및 소프트웨어 테스트 환경을 구현하기 위해서, 가상 머신 상에 실제 플랫폼에 기반하는 소프트웨어 스택을 구축하고, 다

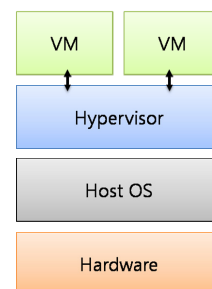
음과 같은 두 가지 기능을 구현하였다.

첫째, 성능 분석을 위한 리소스 모니터(Resource Monitor)를 개발하여, 호스트 머신에서 직접 가상 머신의 상태를 모니터링 할 수 있게 하였다. 둘째, 이벤트 관리를 위한 이벤트 매니저(Event Manager)를 개발하여, 호스트 머신에서 가상 머신의 이벤트를 저장(Save)하거나 재생(Replay)할 수 있게 하였다. 사용자 입력과 같은 임의의 이벤트 재생에 따른 시스템 자원의 상태 및 수행 시간을 통해 소프트웨어의 성능을 분석하고 검증할 수 있는 환경을 제공한다.

본 논문의 구성은 다음과 같다. 2 장에서는 시스템의 구조에 대하여 기술하고, 3 장에서는 구현한 플랫폼과, 세부 기능에 대하여 논한다. 4 장에서는 결론 및 향후 계획에 대해 기술한다.

### 2. 시스템 구조

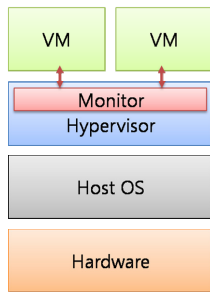
본 논문에서 사용하는 Qemu 는 그림 1 과 같은 가상화 구조를 가진다.



(그림 1) Qemu 가상화 시스템 구조

그림 1 에서와 같이, 호스트 운영체제에서 하이퍼바이저(hypervisor) 를 통해 가상 머신을 구동하게 되는 형태를 가진다.

본 논문에서 제안하는 도구는 하이퍼바이저와 가상 머신 사이의 정보를 획득하기 위해, 그림 2 와 같이 하이퍼바이저의 Monitor 를 통해, 하이퍼바이저와 가상 머신 사이의 정보 교환을 확인할 수 있도록 구현하였다. 즉 하이퍼바이저의 동작 중 원하는 정보를 획득하여, 가상 머신 상에서의 자원을 감시할 수 있게 하였으며, 이벤트에 대한 관리를 가능하게 하였다. 가상 머신은 자신이 실제 시스템에서 독자적으로 구동되는 것처럼 자원을 사용하고, 호스트 머신은 모니터를 통해서 가상 머신의 자원 운영 상태를 확인할 수 있는 것이다.



(그림 2) 가상화 시스템의 Monitor 의 원리

앞에서 언급한 것처럼 이 모니터는 크게 리소스 모니터와 이벤트 매니저로 구성된다. 리소스 모니터를 통해 호스트 머신에서는 가상 머신의 상태로서 CPU, 메모리 등의 자원 사용 양상을 확인할 수 있을 뿐 아니라, 가상 머신 상에서의 각각의 프로세스에 의한 자원의 사용량까지 파악할 수 있다. 이벤트 매니저를 통해서서는 가상 머신에서 특정 이벤트가 발생할 때 이것을 호스트 머신에 전달하는 역할을 수행하며, 이 이벤트를 다시 가상 머신에서 재생되도록 하는 것이 가능하도록 하였다.

이러한 자원 관련 정보나 이벤트 관련 정보들을 호스트 머신 상에서 실시간으로 확인 가능하며, 추후에 호스트 머신에 저장된 내용을 사후 분석에 이용할 수도 있는 구조를 지닌다.

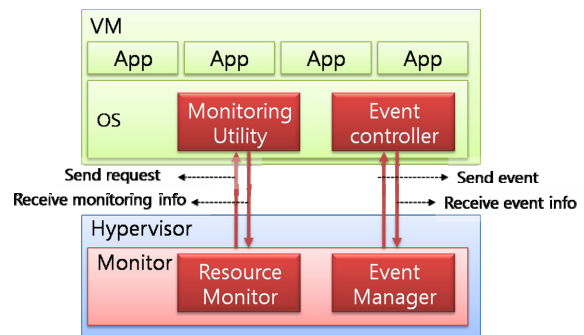
### 3. 구현

본 논문에서 구현한 도구는 가상화 기법을 이용하기 위하여 오픈 소스 기반의 프로세서 에뮬레이터인 Qemu 의 0.9.1 버전을 사용하였다. 그리고 실제 하드웨어 플랫폼과 가장 유사한 가상화 환경을 구축하기 위해, 실존하는 리눅스 기반의 이동 단말 플랫폼인 MCC 보드를 선정하여, 그에 해당하는 Arm 계열의 Mainstone 기반으로 가상화를 구현하였다.

가상 머신을 이용한 편리한 시스템 검증 도구를 구현하기 위해, 리눅스 시스템의 개발을 도와주는 도구들인 Scratchbox 와 BitBake 를 사용하였다. 이것은 커널 및 파일 시스템이 교차 개발 환경을 통해 개발되는 복잡함을 간소화 해준다.

또한, 임베디드 시스템에서 중요시되는 멀티미디어 환경을 테스트 할 수 있는 소프트웨어 스택을 구축하여 가상 머신을 구동할 수 있게 하였다([6], [7]). 리눅스 커널의 2.6.26.3 버전과 Gtk+/DirectFB 기반의 사용자 인터페이스를 사용할 수 있게 하였다([8], [9]).

그리고 사용자 입력을 전달하거나 재생하기 위해서 키 입력 이벤트를 호스트 머신에서 가상 머신으로 전달할 수 있도록 구현하였다. 이것은 Qemu 의 모니터 기능 중에 하나인 Sendkey 를 이용할 수 있으나, Mainstone 의 경우는 USB 키보드를 사용하기 때문에 Sendkey 를 사용할 수 없다. 그래서 USB 키보드를 위한 Usb\_sendkey 기능을 추가로 구현하여, 가상 머신으로 키 입력을 전달할 수 있게 하였다. 그림 3 은 구현한 부분인 리소스 모니터와 이벤트 매니저의 구조를 표현하고 있다. 각각의 기능은 다음과 같다.



(그림 3) 리소스 모니터와 이벤트 매니저

### 3.1 리소스 모니터

가상 머신의 자원들에 대한 상태를 감시하고, 이를 통해 성능을 분석할 수 있는 것이 리소스 모니터이다. 그림 2 에서 알 수 있듯이, 이것은 크게 두 가지의 세부 기능으로 나뉠 수 있다.

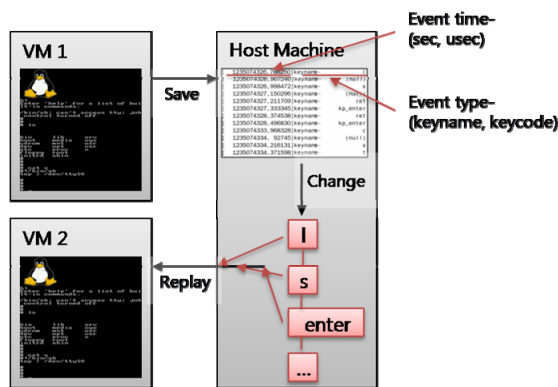
첫 번째는 가상 머신에서 가상화 기법을 통해 동작하고 있는 자원들의 상태를 확인하는 기능이다. CPU, 레지스터, 메모리, 네트워크 장치 등의 상태 정보는 결국 호스트 머신 상에서 하이퍼바이저를 통해 가상화 된 것이기 때문에, 하이퍼바이저에 접근하여 해당 정보를 얻을 수 있다. 호스트 머신의 특정 자원의 정보에 대한 요청이 오면, 해당 자원을 가상화하는 부분을 읽어서 호스트 머신으로 전달하게 된다. 이것은 irq 등의 인터럽트 통계에 대한 정보와, 가상 머신이 사용하는 메모리 블록의 크기 및 사용 양상을 알 수 있으며, 가상 주소와 물리 주소의 실제 내용을 덤핑하여 확인할 수 있는 기능을 포함한다.

두 번째는 가상 머신 상에서 동작하는 프로세스 별 자원 사용량을 파악할 수 있는 기능이다. 가상 머신 상에 리눅스 기반의 시스템의 성능 분석을 수행하는 Top 유틸리티를 구동시켜서 그 결과를 호스트 머신으로 전달하는 것이다. 요청이 발생하면, Top 유틸리티의 결과를 네트워크 장치 또는 시리얼 장치를 이용하여 호스트 머신으로 전송하게 된다. 이 결과로 호스트 머신 상에서 가상 머신의 CPU 사용률과 메모리

사용률, 그리고 각각 프로세스들의 CPU 사용률과 메모리 사용률 및 PID, 실행 경로 등의 세부 정보까지 실시간으로 확인할 수 있게 된다.

### 3.2 이벤트 매니저

호스트 머신에서 가상 머신으로 임의의 이벤트를 전달하거나, 가상 머신에서 발생한 이벤트 정보를 호스트 머신으로 전달하는 기능을 수행하는 것이 이벤트 매니저이다. 그림 2 에서도 알 수 있듯이, 가상 머신 상에서 특정 이벤트가 발생할 때 이벤트를 기록하여 호스트에 전달할 수 있고, 사용자 입력과 같은 I/O 장치를 통한 입력을 호스트 머신으로 전달할 수 있다.



(그림 2) 이벤트 매니저의 재생 기능의 원리

이 기능들을 이용하여, 가상 머신 상에서의 사용자 입력에 대한 정보를 저장하였다가, 다수의 사용자 입력을 임의로 재생할 수 있게 하였다. 그림 3 과 같이 사용자 입력에 대한 정보는 호스트에 저장되어 있기 때문에, 이것을 재생에 이용할 수 있다. 호스트 머신에서 재생을 수행할 경우 이 저장된 정보를 분석하여 이벤트 발생 시간을 고려한 키 입력 스트림을 생성하여, 모니터의 이벤트 매니저를 통해서 가상의 입력을 재현하여 가상 머신으로 전달한다.

사용자 입력의 재생은 같은 가상 머신에서 반복 가능할 뿐 아니라 다른 가상 머신에서도 가능하다. 즉, 같은 사용자 입력에 대한 결과를 커널의 상태에 따라서 수행하여 검증하는데 사용할 수 있고, 다른 가상 머신 상에서의 이벤트 재생의 수행 시간의 확인을 통해 어플리케이션 또는 가상 머신간의 성능을 비교, 분석하는 지표로 활용할 수 있다.

이러한 리소스 모니터와 이벤트 매니저를 통해 특정 시나리오에 따른 사용자 입력에 대한 가상 머신의 상태를 파악할 수 있게 된다. 이것은 가상 머신 상에서 소프트웨어의 검증을 수행하는 용도로 사용될 수 있다.

## 4. 결론

본 논문에서는 호스트 머신에 가상의 임베디드 시스템 환경을 구축하고, 성능 분석 및 검증을 할 수 있는 방법을 제안하였다. 가상 머신의 시스템 자원을 감시하고 이벤트를 관리하는 도구의 개발을 통해, 사

용자 입력에 따른 자원의 상태를 확인할 수 있게 하였다. 이것은 같은 입력 상황에 대한 다른 자원 및 프로세스의 상태에 따른 소프트웨어의 성능을 확인함으로써, 특정 소프트웨어의 성능에 대한 검증의 지표로 사용할 수 있다. 이러한 가상화를 통한 성능 분석 환경은 더욱 복잡해지는 임베디드 시스템 상에서의 소프트웨어의 개발과 수정에 있어 더욱 용이한 기능 검증 환경을 제공할 수 있다.

현재는 구현한 도구를 통한 성능 분석의 검증을 수행하는 자료 및 결과가 부족한 상황이다. 테스트를 통해 결과를 분석해보고, 시스템을 안정화하는 작업이 요구된다.

향후 연구 방향은 다음과 같은 세 가지로 나눌 수 있다. 첫째, 본 도구를 통해 얻은 성능 분석에 관한 정보를 사용자가 쉽게 사용할 수 있게 하는 것이다. 즉, 현재까지 얻은 정보는 사용자가 원하는 성능에 대한 완전한 분석이라고 보기는 어렵다. 가상 머신에 대한 더욱 유용한 정보를 본 도구를 통해 획득하여, 그것을 성능 분석의 결과로 사용자가 이해할 수 있게 보여줄 수 있는 방법이 필요하다. 둘째, 모든 하드웨어의 가상화를 통한 실제와 완전히 유사한 검증 환경을 구축하는 것이다. 현재 모든 하드웨어의 가상화를 제공하지 않기 때문에 실제와 동일한 가상화 환경을 통한 성능 분석과 검증에는 한계가 있다. 셋째, 사용자 입력과 같은 이벤트의 재생에 있어서, 입력 재생을 처리하기 위한 지연시간이 발생하게 되는데, 이것을 조절할 수 있는 효율적 방법을 모색해야 할 것이다.

## 참고문헌

- [1] Fabrice Bellard. Qemu, a fast and portable dynamic translator. In USENIX 2005 Annual Technical Conference, Anaheim, CA, USA, April 2005.
- [2] Geoffroy Vallee, Thomas Naughton, and Stephen L. Scott. System management software for virtual environments. In Proceedings of the 4<sup>th</sup> international conference on computing frontiers, May 2007.
- [3] Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Ian Pratt, Andrew Warfield, Paul Barham, and Rolf Neugebauer. Xen and the art of virtualization. In Proceedings of the ACM Symposium on Operating Systems Principles, October 2003.
- [4] Avi Kivity, Yaniv Kamay, and Dor Laor. Kvm: the Linux Virtual Machine Monitor. In Proceedings of the Linux Symposium, Ottawa, Canada, June 2007
- [5] Mathieu Desnoyers, and Michel R. Dagenais. The LTTng tracer: A low impact performance and behavior monitor for GNU/Linux, In Proceedings of the Linux Symposium, Ottawa, Canada, July 2006
- [6] <http://free-electrons.com/>
- [7] Qemu : <http://bellard.org/qemu/>
- [8] DirectFB : <http://www.directfb.org/>
- [9] Gtk+ : <http://www.gtk.org/>