

OVPsim을 이용한 모바일 컴퓨터의 성능 평가 연구*

양수현, 신민선, 류연승
명지대학교 컴퓨터공학과
e-mail:ysryu@mju.ac.kr

A Study on Performance Evaluation of Mobile Computer using OVPsim

Soo-Hyeon Yang, Min-Sun Shin, Yeonseung Ryu
Dept of Computer Engineering, Myongji University

요 약

최근 컴퓨터 시스템을 소프트웨어적으로 모델링하고 시뮬레이션 성능 평가를 할 수 있는 가상 플랫폼(virtual platform) 기술이 등장하고 있다. OVPsim은 오픈 API를 제공하여 쉽게 프로세서와 주변 장치를 모델링 할 수 있는 가상 플랫폼으로서 소프트웨어 시뮬레이터이지만 매우 빠르게 실행되는 장점을 갖고 있다. 본 논문에서는 OVPsim을 사용하여 간단한 모바일 컴퓨터를 모델링하는 예를 소개하고, 모델링한 컴퓨터 플랫폼에서 리눅스 운영체제를 부팅한 결과와 벤치마크 프로그램을 통해 성능 평가를 수행한 사례를 소개한다.

1. 서론

최근 스마트 폰, 태블릿 PC와 같은 다양한 모바일 컴퓨터의 사용이 매우 빠르게 증가하고 있으며 관련 하드웨어 및 소프트웨어의 기술도 매우 빠르게 발전하고 있다. 특히, 경쟁 제품 간의 차별화를 위하여 고성능의 하드웨어를 장착하는 경쟁이 가속화되고 있다. 2011년에는 듀얼 코어를 장착한 모바일 컴퓨터가 주류를 형성할 것으로 예상되고, 향후 멀티 코어, GPU 등 고성능 프로세서 경쟁이 예고되고 있다. 또한, PRAM, FRAM과 같은 차세대 비휘발성 메모리가 곧 양산될 계획이어서 이를 장착한 모바일 컴퓨터도 등장할 것으로 예상된다. 이와 같이 향후에도 새로운 차세대 하드웨어를 사용하는 스마트 폰이 빠르게 출시될 것으로 예상되고 있다.

모바일 컴퓨터의 하드웨어 기술이 매우 빠르게 발전하고 있지만, 모바일 컴퓨터를 사용하는 사용자들은 성능에 대한 불편함을 느끼는 경우가 많다. 이것은 업체간 경쟁이 심해지면서 모바일 컴퓨터의 성능에 대한 검증이 제대로 수행하지 않은 채 출시 예정일에 맞추어 출시하고 있기 때문이다. 또한, 업체에서도 새로운 하드웨어 설계를 적용한 모바일 컴퓨터의 설계 및 성능을 검증하기가 쉽지 않다.

한편, 최근 컴퓨터 시스템을 소프트웨어적으로 모델링하고 시뮬레이션 성능 평가를 할 수 있는 가상 플랫폼

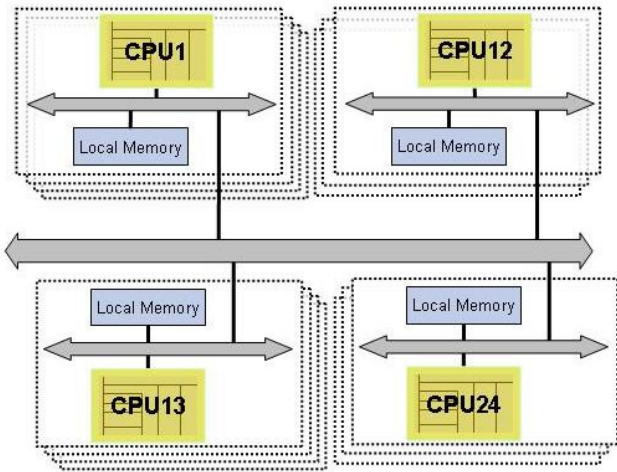
(virtual platform) 기술이 등장하고 있다[1, 2, 3, 4, 5]. SimIt, SkyEye, m5 등은 오픈 소스 시뮬레이터로서 다양한 ARM 아키텍처 버전을 시뮬레이션할 수 있고, m5의 경우는 모듈식의 설계가 장점이다. 최근에는 GPU 시뮬레이션을 위한 연구들이 발표되었다[6, 7, 8]. 이러한 시뮬레이터 중에서 OVPsim은 오픈 API를 제공하여 쉽게 프로세서와 주변 장치를 모델링 할 수 있는 가상 플랫폼으로서 소프트웨어 시뮬레이터이지만 매우 빠르게 실행되는 장점을 갖고 있다[4].

본 논문에서는 OVPsim에서 간단한 컴퓨터를 모델링하는 예를 소개하고, 모델링한 컴퓨터에 리눅스 운영체제를 부팅하고 벤치마크 프로그램을 수행하여 성능 평가를 수행한 사례를 소개한다. 본 논문의 구성은 다음과 같다. 2장에서는 OVPsim의 특징과 구성요소를 설명한다. 3장에서는 OVPsim을 사용하여 컴퓨터 플랫폼을 모델링하는 사례를 간단하게 기술하고 4장에서는 모델링한 플랫폼에서 벤치마크 프로그램을 수행한 성능 데이터를 제시한다. 5장에서 논문의 결론을 맺는다.

2. OVPsim

OVPsim은 Imperas 사에서 만든 개방형 가상 플랫폼(Open Virtual Platform)이다. OVPsim은 웹 사이트에서 무료로 다운로드할 수 있으며[4], 윈도우즈와 리눅스 환경에서 DLL 또는 공유 라이브러리 형태로 설치가 된다. OVPsim은 ARM, MIPS, ARC, NEC v850, openCores OR1K, PowerPC 등 다양한 프로세서를 모델링할 수 있으

* 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (2010-0021897)



(그림 1) OVPsim에서 가능한 매니코어 컴퓨터 모델

며 단일 코어, 멀티 코어, 매니 코어의 모델링도 가능하다. 또한, OVPsim은 다양한 주변장치(peripheral)들을 모델링할 수 있다. 사용자가 C 언어를 사용하여 프로세서를 모델링할 수 있도록 API를 제공한다.

OVPsim은 크게 오픈 소스 모델(open source model), OVPsim 시뮬레이터, 모델링 API의 3가지 구성요소로 이루어진다.

(1) 오픈 소스 모델

OVP는 다양한 프로세서 모델들을 지원하며 미리 컴파일된 오브젝트 코드와 소스 파일을 제공한다. 또한, 24개까지의 멀티 프로세서를 모델링할 수 있다. 또한, RAM, ROM, 캐시, 브리지 등의 시스템 구성요소 및 DMA, UART, 타이머 등의 다양한 주변 장치 모델을 위해 라이브러리를 제공한다. 또한, 이러한 하드웨어 모델에서 수행되는 ucLinux 플랫폼을 제공한다.

(2) OVPsim 시뮬레이터

OVPsim은 OVP 프로세서와 주변장치 모델을 매우 빠르게 수행하는 시뮬레이터로서 현재 윈도우즈 XP SP2와 리눅스 페도라 코어에서 사용할 수 있다. OVPsim은 빠른 실행을 위하여 타겟 명령어를 x86 호스트 명령어로 동적으로 번역하는 Just-in-time 코드 모핑(morphing) 기술을 사용한다. 32 비트 RISC 프로세서에서 OS를 부팅할 때 2-500 MIPS로 수행되고 있다. OVPsim은 디버깅을 위해 GDB 인터페이스를 지원한다.

(3) 모델링 API

하나의 컴퓨터의 모델을 생성하려면, 플랫폼, 프로세서, 주변장치 및 환경을 모델링해야 한다. OVP는 사용자가 모델을 생성할 수 있도록 모델링 API를 제공한다. 이 API는 기본적으로 C 함수로 제공되며, C++ 및 SystemC에서도 사용할 수 있도록 템플레이트를 제공한다. (그림 2)는 모델링 API를 사용하여 간단한 컴퓨터 플랫폼을 생성한 예제 소스를 보여주고 있다. 모델링 API는 여러 API로 구성되어 있다. ICM(Innovative CPU Manager) API는 플랫폼을 생성하고 이를 제어, 연결 및 감시하기 위한 API이

```
int main(int argc, char ** argv) {

    // check for the application program name argument
    if(argc!=2) {
        icmPrintf("%s: expected application name argument\n", argv[0]);
    }

    // initialize CpuManager
    icmInit(0, 0, 0);

    // create a processor
    icmProcessorP processor = icmNewProcessor(
        "cpu1",           // CPU name
        "orlk",          // CPU type
        0,               // CPU cpuId
        0,               // CPU model flags
        32,              // address bits
        model,           // model file
        "modelAttrs",   // morpher attributes
        0,               // enable tracing etc
        0,               // user-defined attributes
        semihosting,    // semi-hosting file
        "modelAttrs"    // semi-hosting attributes
    );

    // load the processor object file
    icmLoadProcessorMemory(processor, argv[1], False, False, True);

    // run simulation
    icmSimulatePlatform();

    // terminate simulation
    icmTerminate();

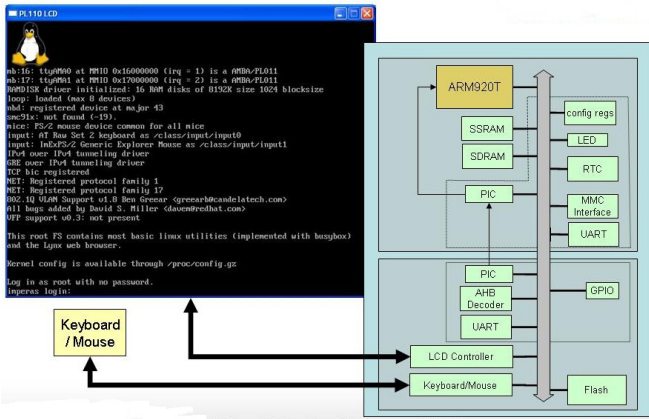
    return 0;
}
```

(그림 2) OVPsim 프로그램 예

다. 예를 들어, 플랫폼에 필요한 멀티 프로세서, 지역 메모리 혹은 공유 메모리, 캐시, 버스 브리지, 어드레스 맵, 인터럽트 등을 생성하고 이들을 연결하는 모델링을 할 수 있다. 또한, 응용 프로그램을 플랫폼의 메모리에 적재시키고 플랫폼을 시뮬레이션할 수 있다. VMI(Virtual Machine Interface) API는 프로세서의 동작(behaviour)을 모델링하기 위한 API이다. 프로세서의 명령어 해독, 실행 등을 모델링한다. 또한, MMU, TLB 등을 효율적으로 모델링할 수 있다. PPM(Peripheral Programming Model)과 BHM(Behavioral Hardware Modelling) API는 주변장치의 동작을 모델링하기 위한 API이다. 시간, 사건과 병행을 모델링할 수 있는 사건 기반의 스케줄링 메커니즘을 제공한다.

3. 모바일 컴퓨터 모델링

(그림 2)는 OVPsim을 사용하여 ARM 프로세서 기반의 모바일 컴퓨터를 모델링하고, 리눅스 운영체제를 부팅한 화면을 보여주고 있다. 프로세서는 ARM920T를 모델링하였고, 메모리는 RAM, 저장장치는 플래시 메모리를 모델링하였다. 이 외에 LCD, 키보드, 마우스 등 주변장치를 모델링하였다. 루트 파일 시스템과 리눅스 커널은 OVP 사이트에서 제공하는 커널 이미지를 사용하였다. 루트 파일 시스템은 램 디스크에 마운트된다. 시뮬레이터는 OVP의 모델링 API를 사용하여 C 언어로 작성하였다. 구



(그림 3) OVPsim을 이용한 시뮬레이션

현한 시뮬레이터에서 리눅스 운영체제가 성공적으로 부팅되었으며 루트 사용자로 로그인하여 응용 프로그램을 수행할 수 있었다.

4. 성능평가

<표 1>은 ARM 프로세서 기반의 컴퓨터 모델에서 대표적인 벤치마크 프로그램을 실행한 성능 평가 결과를 보여주고 있다. 벤치마크 프로그램으로는 linpack, fibonacci, dhrystone, peakspeed를 사용하였고, 성능 지수(performance index)로는 모의실험을 통해 수행된 명령어(simulated instructions) 수, 모의실험을 통해 수행된 MIPS(simulated MIPS), 실제 수행된 시간(run time)을 측정하였다.

<표 1> ARM 프로세서 모델에서 벤치마크 프로그램의 실행 결과

벤치마크	Simulated Instructions	Simulated MIPS	Run time
linpack	1,119,997,553	427.2	2.62
fibonacci	2,251,401,608	219.9	10.24
dhrystone	1,082,061,362	370.8	2.92
peakspeed	14,000,001,618	847.7	16.52

5. 결론

모바일 컴퓨터의 하드웨어 기술이 매우 빠르게 발전하고 있지만, 새로운 하드웨어 설계를 적용한 모바일 컴퓨터의 설계 및 성능을 검증하기가 쉽지 않다. 최근 컴퓨터 시스템을 소프트웨어적으로 모델링하고 시뮬레이션 성능 평가를 할 수 있는 다양한 가상 플랫폼(virtual platform) 기술이 등장하고 있다. 이러한 시뮬레이터 중에서 OVPsim은 오픈 API를 제공하여 쉽게 프로세서와 주변 장치를 모델링 할 수 있는 가상 플랫폼으로서 소프트웨어 시뮬레이터이지만 매우 빠르게 실행되는 장점을 갖고 있다.

본 논문에서는 OVPsim의 컴퓨터 모델링 방법을 분석하고 간단한 모바일 컴퓨터 구조를 모델링하였다. 또한,

모델링한 컴퓨터에 리눅스 운영체제를 부팅하고 벤치마크 프로그램을 수행하여 성능 평가를 수행하였다. 향후 연구로는 OVPsim API 소스를 분석하고 차세대 메모리를 추가할 수 있도록 수정할 계획이다. PRAM과 같은 차세대 비휘발성 메모리를 사용하는 컴퓨터 구조를 모델링하고 성능평가 연구를 수행할 계획이다.

참고문헌

[1] SimIt, <http://simit-arm.sourceforge.net>
 [2] SkyEye, <http://www.skyeye.org/>
 [3] gpgpu-sim, <http://www.ece.ubc.ca/~aamodt/gpgpu-sim>
 [4] OVPsim, <http://www.ovpworld.org>
 [5] N. Binkert, et. al., "The M5 Simulator: Modeling Networked Systems," IEEE Micro, Vol. 26, No. 4, 2006.
 [6] S. Raghav, et. al., "Scalable Instruction Set Simulator for Thousand-core Architectures Running on GPGPUs," High Performance Computing and Simulation, 2010
 [7] S. Collange, et. al., "Barra: a Parallel Functional Simulator for GPGPU," International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication System, 2010.
 [8] D. Chatterjee, et. al., "Event-driven Gate-level Simulation with GP-GPUs," Design Automation Conference, June, 2009.