

# 가상 SSD 시뮬레이터

김해성<sup>○</sup>, 김주현, 이성진, 원유집

한양대학교

matia@ece.hanyang.ac.kr, createmain@ece.hanyang.ac.kr, james@ece.hanyang.ac.kr,  
yjwon@ece.hanyang.ac.kr

## VSSIM : Virtual SSD Simulator

Haesung Kim<sup>○</sup>, Joohyun Kim, Seongjin Lee, Youjip Won

Department of Electronics and Computer Engineering

Hanyang University

### 요 약

최근 Flash Memory의 기술이 급격하게 발달하면서 이를 기반으로 한 SSD(Solid-state Disk) 또한 컴퓨터 시스템 분야에서 매우 중요한 요소로 자리매김하고 있다. 하지만 Flash Memory의 Erase-before-Write, Block단위의 삭제, Page단위의 읽기, 쓰기과 같은 몇 가지 제약 사항은 시스템 성능 저하의 주요 원인이 된다. 이러한 문제를 해결하고자 Flash Memory를 효율적으로 사용하기 위한 관련 연구들이 활발히 진행되고 있으며, 이것의 성능을 측정하기 위한 다양한 Benchmark 도구들 또한 속속 출시되고 있다. 본 논문에서는 QEMU와 KVM 기반의 가상머신에 SSD Emulator와 FTL(Flash Translation Layer)을 삽입하여 SSD의 동작을 시뮬레이션하고 이를 통해 전체 시스템의 성능평가가 가능하도록 고안된 Virtual SSD Simulator에 대해서 소개하고 실험결과를 보인다.

## 1. INTRODUCTION

차세대 저장장치로 급부상하고 있는 NAND Flash Memory 기반의 SSD(Solid-state Disk)는 비 휘발성, 저전력, 무소음, 빠른 I/O 처리 등의 장점을 가지고 현재 임베디드 시스템뿐만 아니라 범용 컴퓨터 시장까지 그 활용 범위를 넓혀가고 있다. 더불어 SSD의 성능이 중요시 됨에 따라 이를 측정하기 위한 Benchmark 도구들 또한 다양한 모습으로 등장하고 있다. 본 논문에서 소개하고자 하는 VSSIM은 QEMU와 KVM 기반의 가상머신에 SSD Emulator와 FTL을 삽입하여 SSD의 동작을 시뮬레이션 할 수 있도록 고안된 가상 SSD 시뮬레이터이다. VSSIM은 Flash Memory의 특성, SSD Controller의 설계 방법, FTL과 같은 SSD 시스템의 핵심이 되는 요소들을 사용자가 원하는 데로 모델링하여 사용할 수 있어 다양한 환경에서 실험이 가능하며, 이를 통해 전체 시스템의 성능 평가뿐만 아니라 각 요소들이 시스템에 미치는 영향력 또한 분석할 수 있다.

많은 SSD Simulator들이 실험을 위한 워크로드를 입력하기 위해 다른 trace 유틸리티를 사용하거나 사용자가 임의로 입력하는 경우가 많다. 하지만 VSSIM은 가상머신으로부터 실시간 워크로드를 받아 사용하며, 실험이 일회성으로

끝나지 않고 시스템의 상태를 계속 유지한다. 즉 실제로 SSD를 사용하는 시스템 환경과 완전히 동일한 환경에서 실험할 수 있음을 뜻한다. SSD의 성능에 가장 큰 병목현상이 되는 부분은 overwrite시 발생하는 merge를 수행하는데 드는 비용이다. 현재 디스크의 사용량도 이것에 큰 영향을 미치는데, 기존의 Simulator들은 이러한 사항을 무시하고 디스크가 비어있다는 가정하에 I/O명령을 수행한다. 하지만 이와 같은 동작의 결과는 실제 시스템에서의 동작과 상이한 결과를 낼 수 있다. 반면에 실제 시스템과 동일한 환경을 지원하는 VSSIM의 동작 방식은 시뮬레이션결과에 대한 정확성과 신뢰성을 증명한다. VSSIM의 이러한 장점들은 현재 Flash Memory기반 SSD의 성능개선을 위해 진행중인 다양한 연구들에게 유용한 도구로서 많은 기여를 할 것으로 기대된다.

본 논문의 2장에서는 관련 연구에 대해서 소개하고, 3장에서는 VSSIM의 구조에 대해 설명한다. 그리고 4장에서는 VSSIM의 기능에 대해 설명하고, 이후 5장에서는 VSSIM을 이용한 간단한 실험 결과를 보인 뒤, 마지막 6장에서는 결론을 내린다.

## 2. RELATED WORK

본 장에서는 SSD Simulator 및 성능 분석과 관련된 몇몇 연구들에 대해 소개한다.

Flash-DBSim[1]은 Flash-aware 알고리즘의 성능을 평가하기 위해 유연성 있는 시뮬레이션 환경을 제공하는 도구이다. 대부분의 시뮬레이션 도구들은 특정한 실험환경으로 설정되어있고, garbage collection이나 wear-leveling 정책과 같은 Flash-aware 알고리즘들이 SSD 내부에 통합된 형태로 설계되어 있어 다른 사용자들이 이를 재사용하기가 어렵다. 하지만 Flash-DBSim은 다양한 형태의 Flash-disk를 제공하며, 번거로운 작업 없이 조금의 코드를 수정하는 것만으로 많은 Flash-aware 알고리즘의 각각의 성능을 평가할 수 있다.

CPS-SIM[2]은 clock의 정확도를 이용하여 SSD 내부 동작을 시뮬레이션하는 clock precision SSD simulator이다. 정확한 타이밍과 CPS-SIM의 이용결과로부터 SSD의 bus와 칩의 개수 그리고 이들의 내부 연결을 포함하는 최적의 하드웨어 구성을 찾아낼 수 있다. 또한 이것은 SSD에 최적화된 FTL을 수행하는 Firmware에 대한 빠른 검증과 개발을 할 수 있도록 한다. FTL은 여러 개의 칩과 버스를 동시에 사용하는데, CPS-SIM 이런 동시성을 지원하여 SSD firmware 설계함에 있어 유연성 있는 환경을 제공한다.

Kaoutar et al.이 발표한 “Modeling and Simulating Flash based Solid-state Disks for Operating Systems” [3]에서는 NAND Flash Memory기반의 throughput model device를 제안하고 이를 시뮬레이터 개발에 사용한다. 시뮬레이터에서 사용하는 파라미터 값을 어떤 SSD에서라도 추출하여 사용할 수 있다는 점은 다른 시뮬레이터보다 뛰어나다고 할 수 있다. 또한 커널의 확장으로 실행중인 운영체제에서 페이징 디바이스로 구성할 수 있어 다른 trace나 input없이 실행할 수 있다.

앞서 소개한 시뮬레이터들과 본 논문에서 소개하는 VSSIM의 가장 큰 차이 점은 가상머신 기반의 환경에서 실시간 워크로드를 통하여 실험을 하기 때문에 단순히 SSD의 하드웨어 성능만을 측정하는 것이 아니라 전체 시스템의 성능을 평가할 수 있다는 점이다. 이것을 통해 실험에 사용된 설계의 방법론이 어떤 컴퓨터 시스템에 가장 적합한지도 알 수 있다.

## 3. VSSIM의 구조

본 장에서는 VSSIM을 구현하는데 사용된

가상머신 도구들에 대해 소개하고 SSD의 상태를 실시간으로 확인하는 Monitoring 프로그램에 대해서 설명한다.

### 3.1 Component of Virtual Machine

#### 3.1.1 QEMU

QEMU[4]는 2005년 Fabrice Bellard가 발표한 머신 에뮬레이터로서 타겟 운영체제와 어플리케이션을 수정하지 않고 가상머신상에서 동작할 수 있도록 한다. Windows, Linux, Mac OS X와 같은 여러 호스트 운영체제 시스템 상에서 작동하며 호스트와 타겟 시스템은 서로 다를 수 있다. 주로 하나의 운영체제에서 또 다른 운영체제를 구동하기 위해 사용되며 수행도중 쉽게 멈출 수 있고 시스템의 상태를 검사하거나 저장, 복구 할 수 있어 디버깅을 위해서도 사용된다. QEMU는 CPU emulator, Emulated device, Generic device, Machine description와 같은 서브시스템으로 구성되며 그 외 Debugger와 User interface등을 포함한다.

QEMU는 동적 변환기를 사용하는데, 동적 변환기는 실행 중(runtime) 타겟 CPU의 명령을 호스트 명령으로 변환하고, 변환된 바이너리코드는 나중에 재사용할 수 있도록 변환 캐쉬에 저장한다. 이러한 방식은 한번에 하나의 명령만 fetch, decode하는 인터프리터방식에 비해 더 효율적이다.

#### 3.1.2 QCOW

QCOW는 QEMU가 지원하는 가상 디스크의 이미지 포맷 중 하나이다. Copy-on-Write를 지원하기 때문에 디스크 이미지의 크기는 디스크를 표현하는 물리 파일 디스크와 크기가 같을 필요가 없고, 디스크 이미지의 사용량에 따라 그 크기가 결정된다. Zlib 기반의 압축방식을 사용하며, 디스크의 상태를 기록하는 스냅샷 등의 기능을 제공한다. qemu-img라는 디스크 생성 유틸리티를 통해 qcow 포맷의 디스크 이미지를 생성하여 사용할 수 있다.

#### 3.1.3 KVM(Kernel-based Virtual Machine)

KVM[5]은 리눅스 커널 2.6.20이상의 버전에서 지원하는 가상화 해법이다. 커널은 가상화 유틸리티인 kvm을 사용함으로써 커널 자체가 hypervisor로 동작할 수 있게 만든다. Intel VT-X나 AMD-V와 같은 가상화 기술을 지원하는 하드웨어 플랫폼에서는 전체 가상화(Full virtualization)를 지원한다.

그림 1에 보인 것처럼 KVM 기반의 모든 가상머신은 리눅스의 표준 스케줄러에 의해 스케줄링되

는 완전한 리눅스 프로세스로서 동작한다. 일반적인 리눅스 프로세스는 user mode와 kernel mode, 두 가지의 모드를 가지지만 KVM은 guest mode를 추가로 가지는데 이는 자신만의 user mode와 kernel 모드를 가진다.

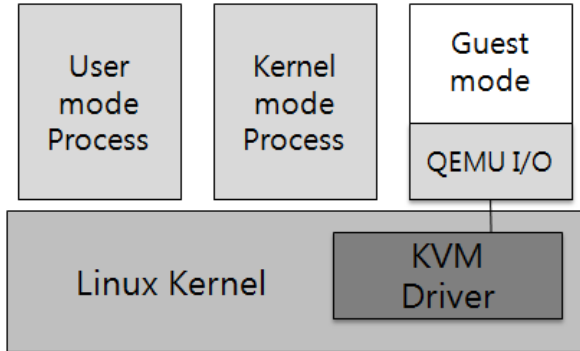


그림 1. Guest mode of KVM

이 Guest Mode 프로세스 상에서 QEMU를 이용한 가상머신이 동작하게 된다. QEMU는 가상머신상에서 동작하는 Guest OS의 I/O를 가로채어 앞서 설명한 것처럼 동적 변환기를 통해 호스트 시스템에 적합한 명령으로 변환하여 전송한다.

### 3.2 SSD Monitoring Tool

SSD Monitoring Tool은 VSSIM에서 SSD의 동작 현황을 실시간으로 확인할 목적으로 만든 보조 유틸리티이다. SSD Monitoring Tool은 SSD Emulator에서 수행되고 있는 명령들의 정보를 소켓을 이용하여 실시간으로 넘겨 받는다. 넘겨받은 정보를 이용하여 Read/Write 요청 횟수 및 처리 속도, Merge 수행 횟수, I/O Amplification 등의 현재 진행 상황을 실시간으로 보여준다.

Monitoring Tool을 사용함으로써 VSSIM이 시뮬레이션 하는 중에도 속도 측정 등의 간단한 분석이 가능하며, 시뮬레이션 종료 후에도 얼마나 많은 양의 데이터를 처리했는지 한눈에 알아볼 수 있다.

## 4. FUNCTION OF VSSIM

VSSIM은 다양한 시뮬레이션 환경을 제공하기 위해 Flash Memory chip의 특성이나, SSD Controller의 설계에 핵심이 되는 부분을 손쉽게 조절하여 사용할 수 있도록 한다. 본 장에서는 이러한 VSSIM의 기능들에 대해 자세하게 살펴본다.

### 4.1 Characteristics of Flash Memory

SSD가 차세대 저장장치로서 주목을 받으면서 그것에 사용되는 Flash Memory의 종류도 매우 다

양해지고 있다. 따라서 VSSIM은 SSD에 사용되는 다양한 Flash Memory Chip의 물리적 특성을 적용시킬 수 있도록 설계되었다.

Features	Specification
Capacity	16384 Mbits
Write Cycle Time	30 ns
Read Cycle Time	30 ns
Program page	800 $\mu$ s
Read Page	60 $\mu$ s
Erase Block	1.5 ms
Flash Size	8192 blocks
Block Size	128 pages
Page Size	2048 Bytes
Max. Read Speed	28.8MB/s
Max. Write Speed	2.4MB/s

표 1 Specification of K9LAG08U0M

표 1은 Flash Memory Chip ‘K9LAG08U0M [6]’의 명세로서, VSSIM에 적용될 수 있는 Flash Memory Chip 특성들의 예를 보여준다. 표에 명시된 시간적 특성과 속도는 SSD Emulator를 구현할 때 중요한 파라미터로 사용되며, Flash, Block, Page Size 또한 FTL 설계 시 고려되어야 할 요소이다.

### 4.2 SSD Controller

SSD Controller의 동작 특성을 결정 짓는 중요한 요소로는 Multi-channels, Multi-ways, FTL이 있다. VSSIM은 Flash Memory뿐만 아니라 SSD Controller 또한 사용자가 원하는 방향으로 설계하여 사용이 가능하다. Channel은 각각의 버스로 연결되어 있어 동시에 접근 가능한 Chip의 개수를 의미하며, way는 하나의 버스에 연결된 Chip의 개수를 말한다. 이는 SSD의 병렬처리 능력을 나타내는 척도라 할 수 있으며, 구성하고자 하는 실험환경에 사용될 Flash Memory chip의 개수를 고려하여 임의로 설정하여 사용할 수 있다. SSD가 동작하는데 가장 핵심이 되는 FTL 또한 다양하게 적용할 수 있다. 여러 FTL을 VSSIM에 적용시켜 시뮬레이션 함으로써 각 FTL들의 성능을 비교하는 데에도 용이하게 사용할 수 있다.

## 5. EXPERIMENT

본 장에서는 VSSIM을 이용한 간단한 실험에 대한 결과를 보여준다. 삼성의 K9LAG08U0M Flash Memory를 사용하고, 이를 4 Channels, 2 Ways-Interleaving 방식으로 설계하여 실험환경을

구축하였다. 표 2는 Hybrid-mapping(FAST[7]) FTL을 사용하여 Windows 7을 설치한 뒤 얻은 실험결과이다. TRIM Command[8]처리 기능의 활성화 여부에 따라 두 가지 결과를 보여주고 있다. Available Blocks는 윈도우 7의 설치가 끝난 뒤 실제 사용하는 데이터가 기록된 Block의 수이며, Written Blocks는 Sector를 기록하기 위해 사용된 Block의 수이다. Written Sectors는 실제로 기록된 sector의 수이며, Erase Avg. Count는 디스크내의 전체 블록에 대한 Erase연산의 평균 값을 계산한 것이다. 마지막 Trim Effect는 Trim Command 처리 기능을 활성화 했을 때 얻은 I/O효율을 sector 단위로 나타내었다.

	TRIM	without TRIM
Available Blocks	29880	30044
Written Blocks	53119	53655
Written Sectors	11754838	11882823
Erase Avg. Count	1.0595	1.0763
Trim Effect	54891	0

표 2. Result of OS Installation

## 6. CONCLUSION

본 논문에서는 QEMU와 KVM을 이용한 가상머신을 기반으로 한 SSD Simulator에 대해 소개 하였다. 실험환경의 유연성과 다양한 방면으로 성능 분석이 가능함은 많은 연구분야에 있어서 높은 활용도를 보인다. 하지만 최근 출시되는 Flash Memory나 SSD는 해당 기관에서 설계 세부사항에 대한 정보를 공개하지 않아 특정 SSD를 모델링하여 시뮬레이션하는 것에는 어려움이 따른다. 이러한 문제는 해당분야의 기술발전을 위해 기업 또는 각 연구기관들이 해결해야 할 문제이다. 그리고 다양한 FTL을 적용한 실험환경을 구축하기 위해서는 해당 FTL을 직접 구현해서 사용해야 한다. 하지만 현재까지 알려진 FTL만해도 매우 다양하고, FTL에 대한 많은 연구가 진행 중에 있는데 모든 FTL을 직접 구현하여 사용하는 것은 시간적 비용 부담이 매우 커질 수 밖에 없다. 이런 부담을 줄이기 위해서는 FTL 내부에서 공통적으로 사용하는 세부적인 알고리즘들을 모듈화하여 설계하고 FTL의 핵심 알고리즘은 이 것의 인터페이스를 사용하여 구현할 수 있도록 최대한 단순화해야 할 것이다.

## 7. ACKNOWLEDGE

본 연구는 지식경제부 및 한국산업기술평가 관리원의 IT R&D 프로그램(Large Scale Hyper-MLC SSD Technology Development, No. 10035202)의 일환으로 수행하였음.

## 8. REFERENCE

- Jin, P., et al., *A flexible simulation environment for flash-aware algorithms*, in *Proceeding of the 18th ACM conference on Information and knowledge management*. 2009, ACM: Hong Kong, China. p. 2093-2094.
- Lee, J., et al., *CPS-SIM: configurable and accurate clock precision solid state drive simulator*, in *Proceedings of the 2009 ACM symposium on Applied Computing*. 2009, ACM: Honolulu, Hawaii. p. 318-325.
- Maghraoui, K.E., et al., *Modeling and simulating flash based solid-state disks for operating systems*, in *Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering*. 2010, ACM: San Jose, California, USA. p. 15-26.
- Bellard, F., *QEMU, a fast and portable dynamic translator*, in *Proceedings of the annual conference on USENIX Annual Technical Conference*. 2005, USENIX Association: Anaheim, CA. p. 41-46.
- Kivity, A. *kvm: the linux virtual machine monitor*. in *In OLS '07: The 2007 Ottawa Linux Symposium*. 2007.
- NAND Flash Technical Paper, MLC-Large Block, 2/4Gx8 bit, K9HBG08U1M / K9LAG08U0M / K9MCG08U5M*. 2007.
- Lee, S.-W., et al., *A log buffer-based flash translation layer using fully-associative sector translation*. *ACM Trans. Embed. Comput. Syst.*, 2007. 6(3): p. 18.
- Shu, F. and N. Obr, *Data Set Management Commands Proposal for ATA8-ACS2*. 2007, Technical Committee T13.