# SSD 기반의 칩 단위 복구를 위한 Erasure 코드 성능 비교

양유석○ ,허준무, 송재석, 김덕환*
인하대학교 전자공학과
{ysyang, joonmoo, gentlejs82}@iesl.inha.ac.kr, deokhwan@inha.ac.kr

# The performance comparison of Erasure Code to recover from chip units based on SSD

Yu-Seok Yang○, Joon-Moo Huh, Jae-Seok Song, Deok-Hwan Kim*
Dept. of Electronic Engineering Inha University

**요 약**

SSD의 칩을 구성하는 셀의 집적도가 높아지고 시스템 오류가 높아짐에 따라, 저장장치의 고신뢰성을 보장해야 하는 필요성은 점점 증가하고 있다. 현재 SSD 의 오류를 복구하기 위한 방법으로 ECC 가 사용되고 있으나 비트단위의 오류복구만 가능하여 복구 성능의 한계를 가지고 있다. 본 논문에서는 SSD 계층 구조인 하나의 칩을 디스크로 설정하고 내부에 RAID 시스템을 구성하여 칩 단위의 오류 복구 기법을 제안한다. 제안된 칩 단위의 오류 복구 기법은 기존 하드디스크 기반 RAID 시스템에 적용 되었던 세 가지 Erasure 코드들을 SSD 에 적용한 방법이다. 또한 실험을 통해 각각의 Erasure 코드 성능을 비교하였다.

## 1. 서 론

SSD 는 Solid State Drive 혹은 Solid State Disk 의 약어로 일반적인 하드디스크와는 달리 낸드 플래시 메모리를 이용하여 정보를 저장 하는 장치를 말한다. SSD 의 경우 대용량이 요구되므로 적게는 한두 개에서 많게는 수십 개의 메모리칩으로 구성되어진다[1].

최근 SSD 는 플래시 메모리 가격의 하락에 힘입어 일반 소비자 시장으로 확대되고 있고 IDC(International Data Corporation) 발표에 따르면, 경기 침체 및 IT 지출 감소에도 불구하고, 2009년 SSD 시장은 출하량 규모가 1,100만대를 넘어서며 전년대비 14% 증가하였고 IDC는 앞으로도 SSD 도입이 증가하며 2010년을 포함해 이후 지속적으로 연평균 54%씩 성장할 것으로 보인다고 예상하고 있다.

플래시 메모리 가격 하락의 가장 큰 원인으로 기존의 SLC(Single Level Cell) 방식에서 MLC(Multi Level Cell) 방식으로 전환되면서 같은 칩을 사용하여도 두 배의 저장 공간을 가질 수 있다. 그리고 생산 원가가 떨어지고 시장 가격이 떨어지면서 더욱 많은 수요가 발생하기 때문에 대량 생산으로 인해 가격이 하락하는 요인이 되었다. 최근 2009년 4분기에는 미국의 마이크론과 인텔이 공동으로 1개의 메모리에 3 비트를 저장할 수 있는 3bit/셀 방식의 낸드 플래시를 개발을 완료하였다[2].

하지만 낸드 플래시 메모리는 마모도로 인한 수명제한(life time)을 가진다. SLC 의 경우 읽고 쓰기 가능 횟수가 약 10만회인 것에 비해 MLC 의 경우는 1만회 정도로 SLC 의 1/10 정도로 감소하며, 앞으로 셀에 저장할 수 있는 비트 수 가 증가함에 따라 재기록 가능 횟수는 더욱 감소 될 수밖에 없다.

이러한 제한적인 단점을 가지는 스토리지 시장에서 SSD 가 하드디스크와의 경쟁력을 올리기 위해서는 플래시 메모리 셀의 집적도를 올려 단가를 낮추는 방법이 사용될 수밖에 없다. 집적도가 높아짐으로 생기는 소자간의 간섭으로 인한 오류를 해결하기 위해서는 데이터의 복구 기능을 강화해야 한다.

현재 플래시 메모리의 일반적인 복구 방법으로는 ECC(Error Correction Code) 를 사용하고 있다. ECC 는 비트 단위로 검출 과 복구하는 방법을 제공하지만 칩 단위의 손상과 같은 불규칙한 오류에 대해서는 복구 할 수 있는 방법이 없으며, 이러한 오류 복구 기술의 연구도 미비한 상태이다.

본 논문에서는 SSD 내부에 존재하는 다수의 칩을 하나의 디스크로 가정하고, 데이터를 분할하여 저장하는 기술인 RAID(Redundant Array of Independent Disks) 시스템에 적용하여 칩 손상과 같은 불규칙한 오류에 대해 SSD 기반의 칩 단위의 복구 방법을 제안한다.

본 논문의 실험을 통해 SSD 내부의 존재하는 칩 기반의 RAID 시스템을 구현하였다. 또한, 높은 복구 성능을

* 교신저자, 인하대학교
※ 이 논문은 2010년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (KRF-2010-0008355).
※ 본 논문은 지식경제부와 한국산업기술 재단의 전략기술인력양성 사업으로 수행된 연구결과임.

가지는 Erasure 코드를 찾기 위해 다양한 Erasure 코드들을 구현된 RAID 시스템에 적용하여 복구 성능을 비교 분석한다.

본 논문의 구성은 다음과 같다. 2장은 SSD 의 구성과 현재 사용하고 있는 에러 복구 기술인 ECC 에 대하여 설명한다. 3장은 논문에서 제안하는 SSD 칩 단위의 복구방법과 Erasure 코드에 대해 설명한다. 그리고 4장에서 실험과 성능평가를 하고, 마지막으로 5장에서 본 연구의 결론을 낸다.

## 2. 관련 연구

### 2.1 SSD의 구성

그림1은 일반적인 운영체제에서의 저장매체별 접근 방법을 나타낸다. SSD 는 일반 낸드 플래시가 가지고 있는 여러 특성들을 보안하기 위해 사용하는 FTL(Flash Translation Layer) 을 컨트롤러 칩으로 제조하여 내부에 하드웨어로 구성된 디스크이다. 낸드 플래시의 특성을 숨기고 다른 디바이스 계층 없이도 일반 하드디스크와 같이 파일 시스템에서 바로 접근 할 수 있다.

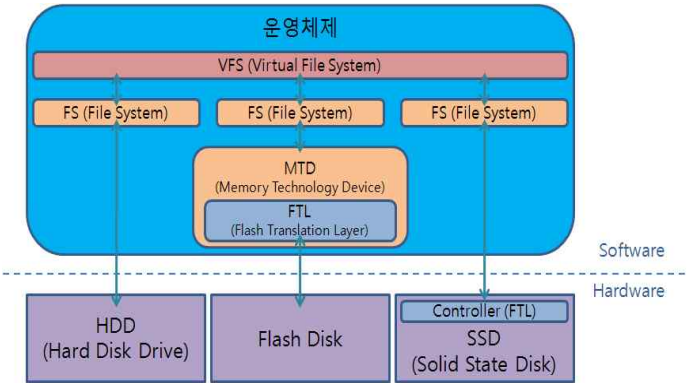


그림 1 운영체제에서의 저장매체별 접근 방법

그림 2는 Mtron 사의 SSD 내부의 구조를 보여준다[3]. 위와 같은 구조를 살펴보면 PC와의 데이터를 주고받기 위한 ATA (Advanced Technology Attachment) 호스트 인터페이스와, 읽기와 쓰기 접근을 빠르게 하기 위해 캐시 버퍼로 SDRAM 을 사용하고 있으며, 마모도 표준화를 위한 FTL 의 연산을 담당하는 ARM7 CPU 와 통신을 위한 AMBA (Advanced Microcontroller Bus Architecture), 그리고 낸드 플래시 메모리 칩들을 위한 컨트롤러가 존재한다. 낸드 플래시 메모리는 병렬로 구성되어 있다.

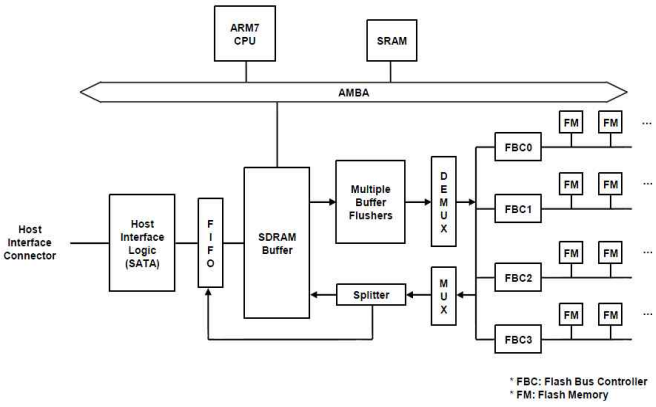


그림 2 SSD 내부 구성도

### 2.2 ECC (Error Correction Code)

낸드 플래시 메모리는 전기적으로 데이터를 기록하는 기억장치이기 때문에, 읽기/쓰기 작업 도중 1비트 또는 2비트가 소자에 잘 못 기록되거나 읽히는 비트 플립핑(bit flipping)이 발생하기 쉽다. 이러한 비트 플립핑은 경우에 따라서 시스템이 심각한 오류를 발생 시킬 수 있는데, 이러한 문제를 방지하기 위해 낸드 플래시 메모리는 ECC를 사용하여 오류를 검출하고 정정한다[4].

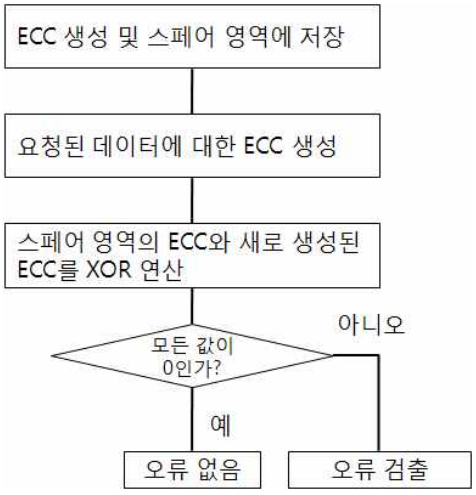


그림 3 ECC 검사 절차

그림 3은 ECC 검사 절차를 나타내고 있다. 먼저 쓰기 요청이 들어오는 동시에 ECC 가 생성되어 스페어 영역에 저장되고, 읽기 요청이 있을 경우 그 부분에 대한 ECC 를 생성하여 스페어 영역에 있는 ECC 와 XOR 연산을 하여, 그 값이 0으로 일치 할 경우 오류가 없는 것으로 판단하고, 아닐 경우 오류를 검출하여 수정하는 방식의 절차를 가진다[5].

ECC 는 검출 및 정정 능력과 연산 방법에 따라 다양

하게 존재한다. SLC 구조의 NAND 플래시의 경우 2비트 검출, 1비트 정정 가능한 해밍코드(Hamming code)를 가장 많이 사용하며, MLC 구조의 경우 BCH 코드 같은 4비트 검출, 2비트 정정 가능한 방법을 사용하고 있다[6]. ECC 를 사용할 경우 원본 데이터를 그대로 비교하는 경우보다 적은 용량의 데이터를 비교하게 되므로 검사 속도가 빠르다는 장점이 있지만, 칩 같이 큰 단위의 복구 능력에 한계가 있다는 단점이 있다.

## 3. SSD 칩 단위 복구를 위한 시스템 구현

낸드 플래시 메모리 내부의 셀 집적도가 높아짐으로 인해 발생되는 오류를 ECC 의 사용으로 신뢰성을 높였지만, 칩 하나 이상 손상되었을 경우 같은 불규칙한 오류에 대응하는 복구 방법은 없다. 따라서 본 논문에서는 칩 단위 복구를 위한 시스템을 제안한다.

### 3.1 단일 SSD 기반 RAID 시스템의 설계 및 구현

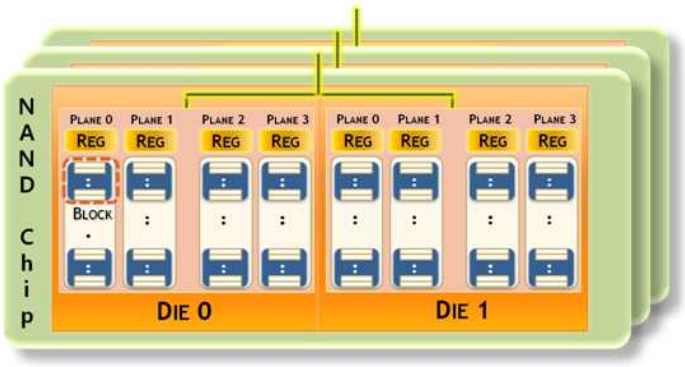


그림 4 다수의 낸드 칩으로 구성된 SSD 구조

SSD는 그림 4 에서 보는 바와 같이 블록, 플레인, 다이의 구조를 갖는 다수의 낸드 칩으로 구성되어 있으며, 병렬 구조를 통해 빠른 입/출력 성능을 갖는다.

본 논문에서 제안하는 단일 SSD 기반 RAID 시스템의 구조는 그림 5와 같다.

파일 시스템에서의 입출력 요청이 들어오면, SSD 내부의 플래시 메모리 컨트롤러를 거치게 된다. 플래시 메모리 컨트롤러는 Erasure 코드가 적용된 RAID-6 시스템과 FTL을 거쳐 각각의 플래시 메모리 칩에 입출력 명령으로 데이터가 저장 된다. RAID는 중복성을 이용하여 칩에 저장하고, 중복된 데이터를 이용하여 오류를 복구하는 방식을 사용한다. 그 중에서도 두 개의 칩의 오류 발생시에도 복구 가능한 RAID-6 시스템을 이용하여 신뢰성을 향상 시킨다.

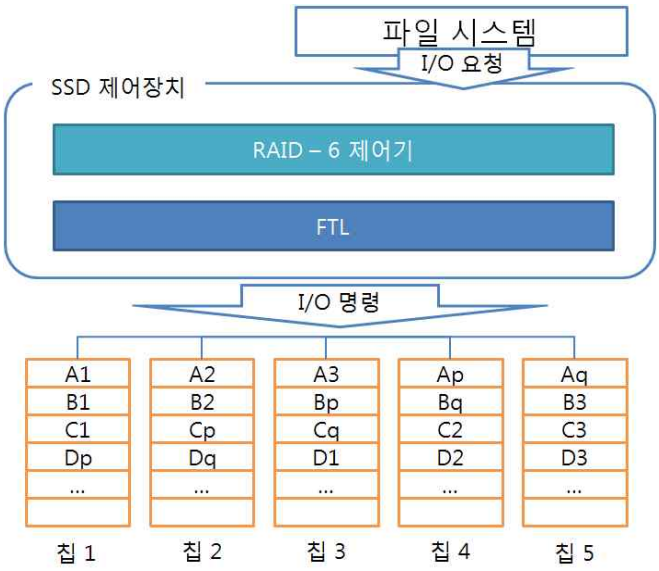


그림 5 단일 SSD 기반 RAID 구조

본 논문에서는 하드디스크 기반 RAID-6 시스템에서 사용된 Reed-Solomon, EVENODD, Liberation, 3가지의 Erasure 코드들을 단일 SSD 칩 기반 RAID-6 시스템에 적용하여 성능을 비교하였다.

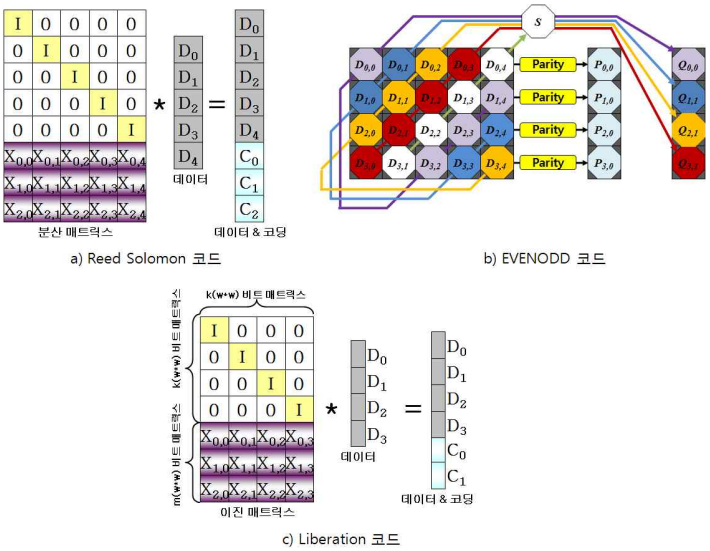


그림 6 Easure 코드

그림 6-a의 Reed-Solomon 코드는 RAID-6 시스템에서 가장 보편적으로 쓰이는 방식이다. Reed-Solomon 코드는 갑작스런 오류에 강하다는 장점이 있으나, 분산 매트릭스를 사용하여 구성되어 있어서 연산 복잡도로 인한 성능저하의 요인이 있다[8].

그림 6-b 의 EVENODD 는 코드의 복잡도를 줄여 단순성에 특징을 가지는 기법이다. EVENODD 기법의 가장

큰 장점은 소수 m개의 데이터 디스크를 사용할 경우에, 단지 두 개의 패리티 디스크만을 추가적으로 사용하므로 패리티 디스크에 대한 오버헤드가 작다는 것이다. 하지만 대각 XOR 값인 S에 대한 오버헤드가 존재하며, 병렬 패리티 기법으로 인한 병목 현상이 발생하는 문제가 있다[8].

그림 6-c 는 James S. Plank이 제안한 Liberation 코드 기법이다[9]. Liberation 코딩은 Reed-Solomon 코드와 유사하지만, 비트 매트릭스를 사용 하여 빠른 연산이 가능하다.

## 4. 실험 및 성능 평가

### 4.1 실험 환경

실제 SSD 내부의 컨트롤러는 하드웨어 칩으로 구성되어 있고 실제로 수정 할 수 없어, 하나의 SSD 를 사용하여 디스크의 칩 단위의 용량으로 파티션을 분할하였다. 칩 단위로 분할한 파티션을 가상의 칩 이라 생각하여 본 논문의 아이디어를 적용하고 Erasure 코드를 사용하여 복구 성능을 비교 하였다. 성능 평가에는 인텔 Core 2 2.13 GHz CPU, 2 GB 메인 메모리와 SSD 로는 Mtron사의 MSD-SATA 32GB 디스크를 사용하였으며, 리눅스 커널 2.6.27에서 Jerasure 라이브러리[11]를 사용하여 실험 하였다.

### 4.2 1개의 칩 오류를 복구 하는 경우

SSD의 복구 성능 측정을 위해 1개의 20MB 텍스트 파일과 서로 용량이 다른 다수 개의 파일들을 실험 데이터로 사용하여 1개의 칩을 복구 하는 경우와 2개의 칩을 복구 하는 경우에 대해 각각 복구 속도(MB/Sec)를 측정하였다.

그림 7 은 1개의 칩이 손상되어 복구 하였을 때의 각 Erasure 코드 별 복구 성능을 나타낸다. 단일 파일의 경우 거의 비슷한 성능을 보여주었고, 단일 보다 복수 파일 때는 연산횟수가 더 많아지기 때문에 분산 매트릭스로 인해 연산 복잡도가 높은 Reed-Solomon 코드보다 EVENODD 가 약 23%, Liberation 코드가 약 28% 정도 좋은 성능을 보였다.

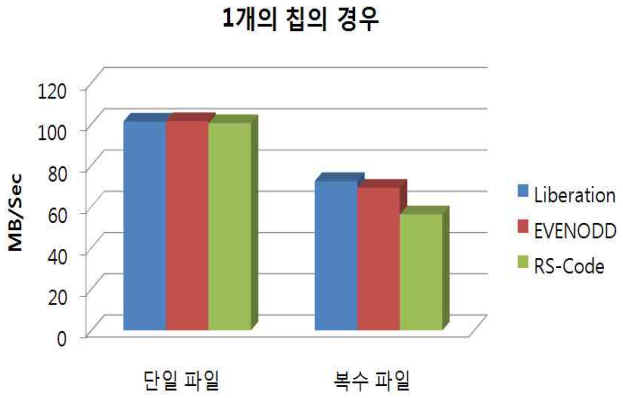


그림 7 1개 칩의 경우

### 4.3 2개의 칩 오류를 복구 하는 경우

그림 8 은 2개의 칩이 손상되어 복구 하였을 때의 복구 성능을 나타낸다.

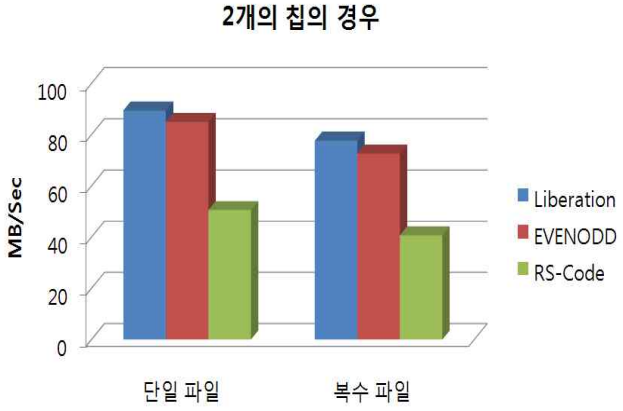


그림 8 2개 칩의 경우

한 개의 칩 때와 같이 단일, 복수 파일 모두 분산 매트릭스 방식의 Reed-Solomon 코드와 비교했을 때 EVENODD 코드가 약 70% 정도, 비트 매트릭스를 사용하는 Liberation 코드가 약 75% 정도의 좋은 성능을 기록했다.

## 5. 결론 및 향후 연구

낸드플래시의 한 셀 당 집적도가 높아지면서 불규칙한 오류가 발생 할 가능성이 높아졌다. 이러한 불규칙한 오류에 대처하기 위한 한 방법으로 칩 단위의 복구 방식을 제안하였다. 그 방법으로는 칩을 하나의 디스크로 지정하여 RAID로 구성하였고, 칩 단위로 복구 할 수 있는

방법을 보였다. 그리고 세 가지 Erasure 코드를 단일 SSD 기반 RAID 시스템에 적용하여, SSD 특성 대비 성능을 분석하고 평가 하였다.

다양한 Erasure 코드들에 대하여 복구 성능을 실험한 결과로, Reed-Solomon 코드는 높은 계산 복잡도의 문제와, EVENODD 코드는 분산 기법을 사용하여 생기는 병목 현상으로 생기는 오버헤드로 인해, 비교적 비트 매트릭스를 사용하여 계산 속도가 빠른 Liberation 코드가 좋은 복구 성능을 보여주었다.

Erasure 코드들이 하드디스크 기반으로 적용되어져 있기 때문에 단일 SSD 기반 RAID 시스템에 적용하였을 경우 데이터가 하나의 칩에 편중되어 마모도가 높아지는 문제가 발생한다. 따라서 향후 SSD 의 특성을 고려한 Erasure 코드를 연구할 계획이다.

## 6. 참고문헌


[1] 정승국, "차세대 스토리지 SSD 기술 동향," 기술동향 통권, 1369호, 2008년.

[2] Micron Technology, Inc, "3-Bits/Cell NAND Flash," M Flash Memory Summit 2009, 2009.

[3] MTRON STORAGE TECHNOLOGY, "MSD-SATA 3525 Product Specification," 2008.

[4] S. Gregori, A. Cabrini, O. Khouri, and G. Torelli, "On-chip error correcting techniques for new-generation Flash memories," Proceedings of the IEEE, 2003.

[5] Samsung Electronics Co. Ltd, "ECC Algorithm (512Byte)"

[6] 이동환, 박광희, 팽소호, 김덕환, "플래시 메모리의 ECC 무결성을 보장하기 위한 IRAW ECC," 한국정보과학회, 2008.

[7] Peter M. Chen , Edward K. Lee , Garth A. Gibson , Randy H. Katz , David A. Patterson, "RAID: high-performance, reliable secondary storage," ACM Computing Surveys (CSUR), 1994.

[8] Mario Blaum, Jim Brady, Jehoshua Bruck, and Jai Menon, "EVENODD: An Efficient Scheme for Tolerating Double Disk Failures in RAID Architectures," IEEE Trans. on Computers, vol. 44, no. 2, Feb, 1995.

[9] James S. Plank, "The RAID-6 Liberation Codes," FAST-2008: 6th Usenix Conference on File and Storage Technologies, San Jose, CA, 2008

[10] 송재석, 허준무, 이동환, 박광희, 김덕환, "Liberation 코드 기법을 적용한 SSD 기반의 RAID-6 시스템 성능비교," 한국정보과학회, 2009.

[11] S. Plank, S. Simmerman and C. D. Schuman, "Jerasure: A Library in C/C++ Facilitating Erasure Coding for Storage Applications," Technical Report CS-08-627, University of Tennessee Department of Electrical Engineering and Computer Science, August, 2008.