

논문 2010-47CI-6-6

# 신뢰성 향상과 성능개선을 위해 다양한 Erasure 코드를 적용한 SSD 기반 RAID-6 시스템 구조

(SSD-based RAID-6 System Architecture  
for Reliability and Performance Enhancement)

송 재 석\*, 허 준 무\*, 양 유 석\*, 김 덕 환\*\*

(Jae-Seok Song, Joonmoo Huh, Yu-Seok Yang, and Deok-Hwan Kim)

## 요 약

전통적인 대용량 스토리지 시스템을 위해 하드디스크 기반의 RAID 시스템이 사용되고 있다. 그러나 하드디스크 기반의 RAID 시스템은 외부의 충격에 약하기 때문에 데이터의 신뢰성이 낮고 스피들 모터의 빈번한 동작으로 인해 전력소모가 많다. 따라서 본 논문에서는 하드디스크를 대체하여 다양한 Erasure 코드를 적용한 SSD 기반 RAID-6 시스템을 제안한다. 제안된 방법으로 Reed-Solomon, EVENODD, Liberation 코딩 기법을 파일시스템 레벨과 디바이스 드라이버 레벨에 각각 적용하였다. 그리고 SSD의 수명저하에 미치는 영향을 최소화하기 위하여 데이터 할당 기법을 적용하였다. 제안된 Erasure 코드가 적용된 RAID-6 시스템에 HDD와 SSD를 사용하여 실험한 결과 Liberation 코드가 다른 코드에 비해 SSD 수명저하에 영향을 적게 끼침을 파악하였으며, Erasure 코드들을 파일시스템 레벨에 적용 하였을 경우보다 디바이스 드라이버 레벨에 적용하였을 때 성능이 더 높았다. 또한 HDD를 기반으로 사용하였을 경우 보다 SSD를 기반으로 사용하였을 경우에 입출력 성능이 4.5%~8.5% 더 향상 되었고 전력소모는 18%~40% 절감되었다.

## Abstract

HDD-based RAID systems have been used in high-capacity storage systems for traditional data server. However, their data reliability are relatively low and they consume lots of power since hard disk drive is weak on shock and its power consumption is high due to frequent spindle motor operation. Therefore, this paper presents new SSD based RAID system architecture using various erasure codes. The proposed method applies Reed-Solomon, EVENODD, and Liberation coding schemes onto file system level and device driver level, respectively. Besides, it uses data allocation method to minimize the side effect of reducing the lifespan of SSD. Detail experimental results show that Liberation code increase wear-leveling rates of SSD based RAID-6 more than other codes. The SSD based RAID system applying erasure codes at the device driver level shows better performance than that at the file system level. I/O performance of RAID-6 system using SSD is 4.5%~8.5% higher than that of using HDD and the power consumption of the RAID system using SSD is 18%~40% less than that of using HDD.

**Keywords :** erasure code, multiple device driver, ssd, raid-6

\* 학생회원, \*\* 정회원, 인하대학교 전자공학과  
(Department of Electronic Engineering,  
Inha University)

※ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (2010-0008355).

※ 본 논문은 지식경제부와 한국산업기술진흥원의 전략기술인력양성사업으로 수행된 연구결과임.

접수일자: 2010년7월16일, 수정완료일: 2010년10월25일

## I. 서 론

전통적인 대용량 서버급 환경에서의 저장장치 시스템(Storage System)으로 여러 개의 디스크를 병렬 구조로 연결한 RAID (Redundant Array of Independent Disks) 시스템이 사용되고 있으며, 하드디스크의 수가

증가할수록 기계적인 동작이 추가적으로 발생되기 때문에 전력소모가 증가하게 될 뿐만 아니라 외부의 충격으로 인해 쉽게 데이터가 손실될 가능성이 높다. 또한 하드디스크는 소음이 많고 발열이 쉽게 발생한다는 단점을 가진다.

RAID 시스템은 본래 적은 용량을 가진 두 개 이상의 디스크들을 하나의 디스크처럼 사용할 수 있는 환경을 의미하며, 크게 다음과 같은 세 가지 작업을 수행할 수 있다<sup>[1]</sup>. 첫째, 여러 디스크가 동시에 동작해 단일 데이터 흐름을 공급하거나 받아들이게 허용하는 방식으로 다수의 디스크에 존재하는 ‘스트라이핑(Stripping)’ 기법으로 성능을 향상시킬 수 있다. 둘째, 다수의 디스크에 존재하는 데이터를 복사하는 ‘미러링(Mirroring)’ 기법이 있어 단일 디스크의 파손과 관련된 위험을 줄일 수 있다. 셋째, 파손된 단일 디스크의 데이터를 복구하기 위한 ‘패리티(parity)’ 기법이 존재한다. 패리티 기반의 다양한 Erasure 코드 기법들<sup>[2~4]</sup>은 계산 방법이 각각 다르기 때문에 저장장치의 성능에 미치는 영향을 잘 파악해야 한다.

최근에는 전통적인 하드디스크와 달리 기계적인 동작이 없고 전력소모가 낮으며 외부의 충격에 강한 SSD(Solid State Disk)가 등장하였다.

SSD는 다수의 낸드 플래시 메모리로 구성되어 있으며, 다음과 같은 두 가지의 특성들을 고려해야 한다<sup>[5]</sup>.

첫째, SSD는 마모도(Wear leveling)라고 불리는 블록 갱신 횟수 제한을 고려한다. 싱글 레벨 셀(Single Level Cell)에서는 약 10만 번 그리고 멀티 레벨 셀(Multi Level Cell)에서는 약 1만 번 정도의 수명을 가지고 있다. SSD는 멀티 레벨 셀로 구성되어 있기 때문에 싱글 레벨 셀보다 수명의 제한이 낮다.

둘째, 기존의 하드디스크와 달리 SSD는 읽기/쓰기/삭제 3개의 연산의 동작으로 구분할 수 있으며, 읽기/쓰기 연산은 페이지(512B~4KB) 단위로 수행되고 삭제 연산은 블록(16KB~256KB) 단위로 수행된다.

이와 같이 SSD는 고성능/저전력의 장점을 가지고 있어 휴대기기에서부터 엔터프라이즈 서버와 같은 고성능 시스템에 이르기까지 하드디스크를 대체할 저장장치로 각광받고 있다.

따라서 본 논문에서는 서버급 환경에서의 저장장치로서 다양한 Erasure 코드를 적용한 다중 SSD 기반 RAID-6 시스템의 두 가지 구조를 제안한다. 제안한 방법은 Reed-Solomon, EVENODD, Liberation 코딩기법

과 같은 다양한 Erasure 코드들을 파일 시스템 레벨(File System Level)과 디바이스 드라이버 레벨(Device Driver Level)에 각각 적용하였다. 그리고 SSD의 수명 저하에 미치는 영향을 최소화하기 위해서 데이터 할당 기법을 적용한다.

본 논문의 실험을 통해 SSD 기반의 RAID-6 시스템 환경에서 제안된 두 가지 형태의 구조에 각각 적용된 Erasure 코드들의 성능을 평가 및 비교 분석하였다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구를 기술한다. III장에서는 다양한 Erasure 코드를 적용한 다중 SSD 기반 RAID-6 시스템의 두 가지 구조를 제안한다. 그리고 제안된 데이터 할당 기법과 Erasure 코드 연산에 따라 SSD 수명저하에 미치는 영향을 기술한다. V장에서는 실험을 통해 신뢰성 측정, 전력 소모, 인코딩의 성능을 비교 분석한다. 마지막으로 VI장에서는 본 논문의 결론을 맺는다.

## II. 관련 연구

본 장에서는 대용량 스토리지 시스템에서 사용되는 패리티 기반의 하드디스크 RAID 시스템과 SSD의 수명증가를 위해 연구된 Diff-RAID 시스템에 대해 설명한다.

### 1. 하드디스크기반의 패리티 RAID 시스템

기존 하드디스크 기반의 대표적인 패리티 RAID 시스템인 RAID-5와 RAID-6에 대한 설명은 다음과 같다.

#### 1.1 RAID-5 시스템

RAID-5는 최소 3개 이상의 디스크가 있어야만 구성이 가능하며, 보통 5개 이상의 디스크로 구성된다. 또한 디스크에 오류가 발생했을 때에는 패리티(Parity)를 이용해서 데이터를 복구할 수 있기 때문에 패리티 RAID 시스템이라고도 불린다<sup>[6]</sup>.

RAID-5와 같은 스트리지 시스템은 서로 독립적으로 동작하므로, 여러 개의 소규모 어플리케이션 요청 사항을 동시에 처리할 수 있으며, 대용량 입/출력 요청을 처리해야 하는 경우에는 모든 디스크가 활성화되어 하나의 작업을 처리하는 장점을 가진다.

하지만 위와 같이 하드디스크 기반으로 구성된 RAID 시스템은 하드디스크의 기계적 특성 때문에 소모되는 전력이 많이 필요하며, 디스크의 공간이 협조하지

는 경우 성능이 저하되는 문제점을 가지게 된다.

### 1.2 RAID-6 시스템

RAID-6 시스템은 2 개의 패리티를 사용함으로써 공간 효율은 RAID-5 보다 약간 떨어지지만, 2 개의 디스크가 동시에 고장 나도 데이터에는 이상이 없도록 하는 방식이다<sup>[7]</sup>. 따라서 RAID-6 시스템은 최소 4 개의 디스크로 그림 1과 같이 구성 되어야 한다.

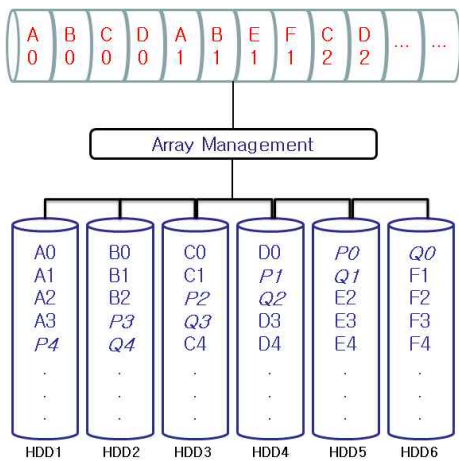


그림 1. 하드디스크 기반의 RAID-6 시스템  
Fig. 1. HDD based RAID-6 System.

따라서 RAID-6 시스템의 장점으로 RAID-5 시스템 보다 데이터에 대한 신뢰도는 좀 더 높아지는 효과를 갖는다. 하지만 RAID-6 시스템의 단점으로 RAID-5 시스템은 패리티를 1 개만 생성하면 되지만, RAID-6 시스템은 패리티를 2 개를 생성해야 하므로, 내부적으로 Erasure 코드를 사용하여 인코딩 할 때, 쓰기(Write) 연산이 복잡해 성능 면에서 RAID-5 보다 약간 떨어진다는 문제점을 가지게 된다.

### 2. Differential RAID

MLC 로 구성된 SSD 는 삭제 연산이 10,000 번으로 제한되어 있다. 따라서 SSD 의 수명 저하로 인한 데이터 손실 및 디스크 오류에 치명적인 단점을 가지게 된다. 최근에는 SSD 기반 RAID 시스템의 연구로 SSD의 수명제한을 고려한 Diff-RAID(Differential RAID) 방식이 연구 되었다<sup>[8]</sup>.

그림 2는 Diff-RAID의 패리티 분산 방법을 표현하였다. 4개의 SSD(SSD0~SSD3) 로 RAID 시스템이 구성되어 있고, 각각의 SSD 로 70 %, 10 %, 10 %, 10 % 패리티 분산(Parity Distribution) 되어 저장된다. SSD0

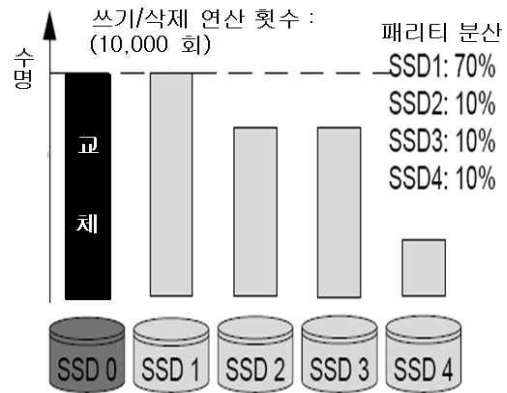


그림 2. Differential RAID 의 패리티 분산에 따른 수명률  
Fig. 2. endurance of parity distribution in Diff-RAID.

의 패리티 분산률이 70%로 가장 높기 때문에 SSD0 로 쓰기/삭제 연산이 집중된다. 따라서, SSD0 가 나머지 SSD (SSD1~SSD3) 보다 낮은 수명률을 보이게 된다.

그리고 SSD0 의 수명저하로 인해 디스크가 오류를 발생하게 된다면 SSD 한 개(SSD4) 를 추가 하여 기존의 RAID 시스템을 계속 유지하게 된다. SSD1은 SSD0 대신에 패리티 분산률이 10 %에서 70 %로 증가 하게 되고 SSD1 로 데이터의 쓰기/삭제 연산이 집중되는 시스템을 보여주게 된다.

위와 같은 Diff-RAID 시스템은 디스크의 수명을 고려한 연구인만큼 SSD 기반의 RAID 시스템의 수명이 효율적으로 관리 되지만 Erasure 코드의 연산으로 인해 SSD 에 미치는 영향에 대해서는 고려되지 않았다.

## III. Erasure 코드가 적용된 다중 SSD 기반 RAID-6 시스템의 구조

본장에서는 Erasure 코드를 파일 시스템 레벨과 디바이스 드라이버 레벨에 적용한 구조를 설명한다. 그리고 SSD 수명저하의 문제점을 고려한 알고리즘과 방법에 대해 설명한다.

### 1. Erasure 코드를 파일 시스템 레벨에 적용한 구조

그림 3은 다양한 Erasure 코드를 적용한 파일 시스템 레벨의 구조를 보여 준다.

하나의 데이터를 Erasure 코드를 통해 파일 시스템 레벨에서 데이터 파일(파일0~파일3)과 패리티 파일(파일p,파일q)로 분할한다. 분할된 파일들은 가상파일 시스템과 MD드라이버를 통해 블록 할당 관리자로 보내어



그림 3. Erasure 코드가 적용된 파일 시스템 레벨 구조  
 Fig. 3. Applying Erasure code in File System Level structure.

진다. 블록 할당 관리자는 파일의 메타데이터 정보를 가지고 섹터 단위(512B~4KB)로 데이터 영역과 패리티 영역을 분류하여 RAID-6 시스템으로 전송한다. 전송된 섹터들은 다중 SSD 기반 RAID-6 시스템에 순차적으로 페이지 단위(512B~4KB)로 저장 된다.

위와 같은 파일 시스템 레벨에 Erasure 코드가 적용된 구조는 데이터를 파일 단위 생성한 후, 섹터 단위로 할당하여 쓰기 연산을 하게 된다. 따라서 SSD 기반의 저장장치에 저장될 때, 생성된 파일들을 페이지 단위로 다시 분할하여 저장하기 때문에 SSD 기반 RAID-6 시스템의 입출력 성능 저하의 요인이 될 수 있다.

2. Erasure 코드를 디바이스 드라이버 레벨에 적용한 구조

Erasure 코드가 적용된 디바이스 드라이버 레벨은 리눅스 커널 소스 내에 존재하는 멀티플 디바이스 드라이버(Multiple Device Driver)의 소스를 수정하여 Erasure 코드를 적용하였으며, 제안한 구조는 그림 4

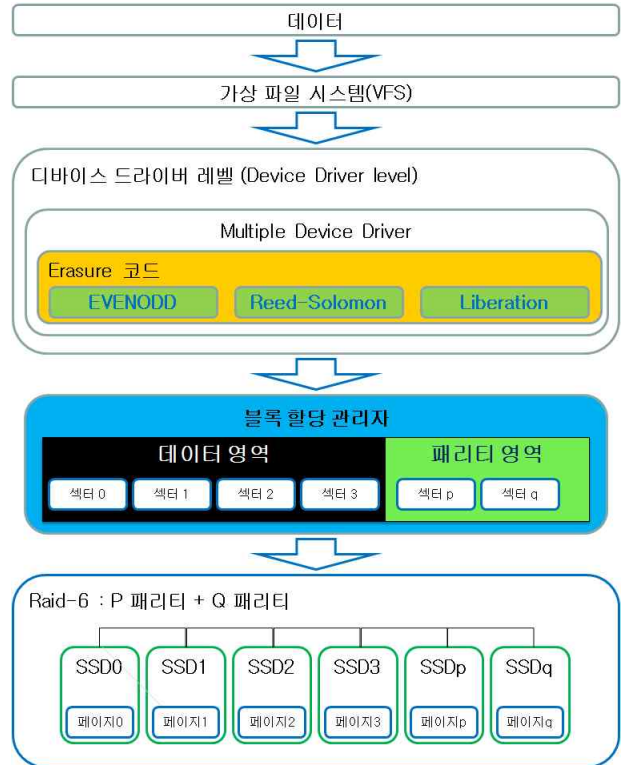


그림 4. Erasure 코드가 적용된 디바이스 드라이버 레벨 구조  
 Fig. 4. Applying Erasure code in Device Driver Level structure.

와 같다.

디바이스 드라이버 레벨에 적용한 구조는 파일 시스템 레벨에 적용한 구조와 달리 하나의 데이터가 가상 파일 시스템을 통과해 MD 드라이버 레벨에 적용된 다양한 Erasure 코드를 통해 인코딩 되어 진다. 그 후, 블록 할당 관리자가 인코딩된 데이터를 데이터 영역과 패리티 영역으로 분할하여 RAID-6 시스템에 순차적으로 저장 한다.

따라서 디바이스 드라이버 레벨에서 Erasure 코드의 인코딩/디코딩 동작이 이루어지기 때문에 SSD 기반 RAID-6 시스템의 입출력 성능이 향상 될 수 있다. 뿐만 아니라 향상된 연산 속도로 인해 전력도 적게 소모 된다.

3. SSD 신뢰성을 고려한 패리티 저장 기법

SSD의 수명저하로 인한 신뢰성 저하의 문제점을 향상시키기 위해 블록 할당 관리 기법을 제안하고 쓰기 횟수를 측정하는 방법을 고려한다.

### 3.1. 블록 할당 관리 기법

표 1은 블록 할당 관리 기법을 알고리즘으로 표현하였다. 블록 할당 관리 기법은 Erasure 코드 연산을 통해 인코딩된 데이터의 메타 데이터(meta data)에 저장된 정보로 분류 할 수 있다. 예를 들어 메타 데이터의 정보가 데이터블록(data block)이면, 블록 할당 관리자 내에서 데이터영역(data area)으로 분류 하게 되어 SSD0 ~ SSD3 로 저장된다. 그리고 메타 데이터의 정보가 패리티블록p(parity block\_p) 이거나 패리티블록 q(parity block\_q)이면, 블록 할당 관리자에서 패리티 영역(parity area)으로 분류 하게 되어 SSDp 와 SSDq 로 분류되어 저장된다.

표 1. 블록 할당 관리 기법  
Table 1. Block allocation Algorithm.

```

알고리즘. 블록 할당 관리 알고리즘
INPUT : encoded block
OUTPUT : sorted block
function block_i/o_allocation_operation
{
    if meta_data information of encoded block is marked
    as data block
    {
        write encoded block sequentially in data area
        of SSD0~SSD3;
        data_count++;
    }
    else if meta_data information of encoded block is
    marked as parity_p block
    {
        write encoded block sequentially in parity_p area
        of SSDp;
        parity_p_count++;
    }
    else if meta_data information of encoded block is
    marked as parity_q block
    {
        write encoded block sequentially in parity_q area
        of SSDq;
        parity_q_count++;
    }
}

function program_erase_operation_counter
{
    if (data_count || parity_p_count || parity_q_count )
    >= 9000
    {
        for (i; i=9000; i++;)
        {
            printk ("<0> your disk life_cycle = %5d", i);
        }
    }
}
    
```

### 3.2. SSD 수명을 고려한 데이터 및 패리티 분류 방법

그림 5는 블록 할당 기법으로 갱신 횟수가 잦은 패리티 블록을 하나의 SSD 로 집중하여 SSD 의 수명을 향



그림 5. SSD 수명을 고려한 인코딩된 블록의 블록 할당 기법

Fig. 5. Block allocation scheme of encoded for considering lifetime of SSD.

상 시키고 데이터와 패리티를 효율적으로 관리 하는 방법을 보여준다. 데이터는 총 4개의 SSD 로 분산되어 관리되며, 패리티p 와 패리티q는 하나의 SSD 에 집중되게 저장하여 다른 SSD 의 수명저하를 최소화 한다.

표 1의 쓰기/삭제 연산 횟수 측정은 MLC 타입의 SSD 수명이 10,000 회로 제한되어 있기 때문에, SSD 의 수명을 확인 할 수 있도록 횟수를 측정한다. 만약 데이터 및 패리티 SSD 의 쓰기/삭제 연산 횟수가 9,000 회가 된다면, 이를 감지해 주어 사용자에게 알려줌으로써 데이터 및 패리티 디스크의 교체 알림 메시지를 출력해 주어 데이터의 보전성을 유지시켜준다.

## 4. 제안된 다중 SSD RAID-6 시스템 구조에 적용된 Erasure 코드 기법

다양한 Erasure 코드들은 인코딩 기법에 따라 계산 방법이 달라지기 때문에 SSD 에 미치는 영향에 따라 성능 및 전력소모가 다르게 측정 된다. 따라서 SSD 환경에 적합한 Erasure 코드를 찾기 위해 다음과 같은 Erasure 코드들을 SSD 기반의 RAID 에 적용하였으며, Erasure 코드들의 특징과 SSD에 미치는 영향은 다음과 같다.

### 4.1. Reed-Solomon 코드 기법

그림 6은 Reed-Solomon 코드가 적용된 SSD 기반의 RAID-6 시스템을 보여준다<sup>[2]</sup>.

Reed-Solomon 코드는 가장 강력하고 보편적인 코딩 기법이다. Reed-Solomon 코딩 기법은 여러 개의 오류를 복잡한 매트릭스 곱으로 복원 할 수 있다. 분산 매



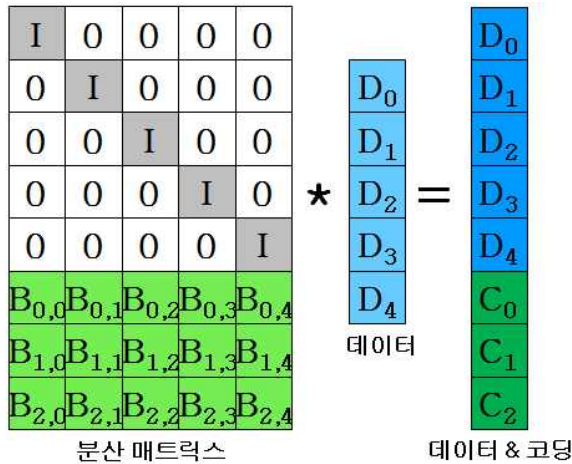


그림 6. Reed-Solomon 코드의 인코딩  
Fig. 6. Encoding using Reed-Solomon Code.

트릭스 B는 매트릭스 I와 하나의 매트릭스로 구성되어 있고 데이터 D와의 곱셈 연산으로 데이터와 코딩 정보를 생성할 수 있다.

하지만 분산 매트릭스기반의 Reed-Solomon 코드 연산은 계산하는데 많은 비용을 필요로 한다. 따라서 SSD 기반 환경에서는 비트 갯수 횡수가 증가함에 따라 SSD 마모도에 미치는 영향이 높아져 시스템의 전체적인 성능이 저하 될 수 있다.

4.2. EVENODD 코드 기법

그림 7은 EVENODD 코드가 적용된 SSD 기반의 RAID-6 시스템을 보여준다<sup>[3]</sup>.

독립적인 두 개의 패리티 열을 가지고 있으며, 두 개의 패리티 열은 각각 데이터들의 체크섬으로 계산 할 수 있다.

첫 번째 패리티(C<sub>0,0</sub>~C<sub>0,3</sub>)는 디스크 행렬의 각 행을 모두 XOR 한 값이며 다음과 같은 식으로 표현 할 수 있다. 식 2 를 살펴보면, j는 열을 의미하며 0 ≤ j ≤ 3 의 모든 열의 XOR 계산 된다.

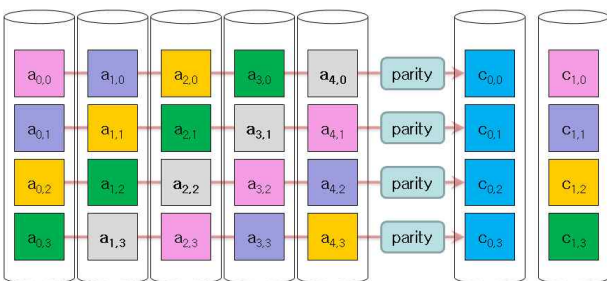


그림 7. EVENODD 코드 인코딩  
Fig. 7. Encoding using EVENODD Code.

$$C_{0j} = a_{0j} \oplus a_{1j} \oplus a_{2j} \oplus a_{3j} \oplus a_{4j} \quad (2)$$

두 번째 패리티는 EVENODD 코드의 특수한 대각 패턴과 신드롬(Syndrome)을 함께 XOR 연산하여 계산된 결과로 식 3 과 같이 표현 할 수 있다.

$$S = a_{4,0} \oplus a_{3,1} \oplus a_{2,2} \oplus a_{1,3} \quad (3)$$

$$C_{1,0} = S \oplus a_{0,0} \oplus a_{2,3} \oplus a_{3,2} \oplus a_{4,1}$$

$$C_{1,1} = S \oplus a_{1,0} \oplus a_{0,1} \oplus a_{3,3} \oplus a_{4,2}$$

$$C_{1,2} = S \oplus a_{2,0} \oplus a_{1,1} \oplus a_{0,2} \oplus a_{4,3}$$

$$C_{1,3} = S \oplus a_{3,0} \oplus a_{2,1} \oplus a_{1,2} \oplus a_{0,3}$$

위와 같이 계산방법으로 EVENODD 코드가 SSD 에 미치는 영향은 단순한 XOR 연산으로 Reed-Solomon 코드보다 연산이 단순하기 때문에 SSD 수명저하에 미치는 영향이 적을 것이다.

4.3. Liberation 코드 기법

그림 8은 Liberation 코드가 적용된 SSD 기반의 RAID-6 시스템을 보여준다<sup>[4]</sup>.

Liberation 코드는 Reed-Solomon 코드와 매우 유사하게 계산 되며, Reed-Solomon 코드에서 사용되는 분산 매트릭스와 달리 이진 비트 매트릭스로 분할하여 데이터와 결합하는 코딩 기법을 사용한다.

일반적으로 Liberation 코드는 k개의 데이터 디바이스와 m개의 코딩 디바이스, 그리고 각각의 w로 고정된 비트로 구성된다. 또한 모든 요소들이 바이너리(binary)로 구성되어 있기 때문에 BDM(Binary Distribution Matrix)이라 불리며, Reed-Solomon 코딩 보다 빠른 연산이 가능하다.

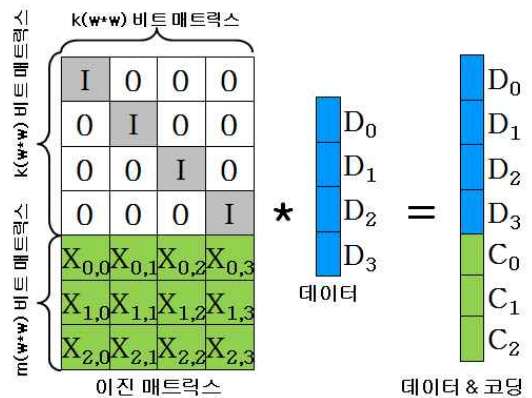


그림 8. Liberation 코드의 인코딩  
Fig. 8. Encoding using Liberation Code.

Liberation 코드는 Reed-solomon 코드와 연산 방법이 매우 유사하지만, 이진 매트릭스를 사용하여 계산되기 때문에 계산 복잡도가 낮으며 빠른 연산을 할 수 있다. 따라서 SSD 기반 환경에 적용 하였을 때 Reed-Solomon 보다 낮은 비트 갱신 횟수로 인해 SSD 마모도에 미치는 영향과 SSD 수명저하에 미치는 영향이 적다는 특징을 가지고 있다.

### V. 실험 및 성능 평가

본 논문의 실험은 SSD 기반의 RAID-6 시스템에서 Erasure 코드들의 연산 방법에 따라 달라지는 마모도를 측정하기 위해 비트 갱신횟수를 측정하여 비교 분석 하였다. 그리고 구현된 하드디스크 기반의 RAID-6 시스템과 세 가지 Erasure 코드가 적용된 SSD 기반 RAID-6 시스템에서의 인코딩 성능과 입/출력 성능 그리고 전력소모에 대하여 측정 및 비교 분석 하였다.

#### 1. 실험 환경

표 2는 본문의 실험을 위해 사용된 실험 환경이다. RAID-6 시스템은 소프트웨어 RAID로 구현하였으며, mdadm 이라는 툴을 사용하였다. 파일 시스템 레벨에 적용된 Erasure 코드들은 Jerasure 코드를 사용하여 SSD 기반의 Erasure 코드에 따른 마모도를 측정하였고, MD드라이버에 적용된 Erasure 코드들은 EVENODD, Liberation 코드인 경우 새로이 구현 하였으며, Reed-Solomon 코드는 커널에서 제공하는 모듈을 수정하여 사용하였다. 그리고 저장장치 시스템의 성능 평가를 위해서 SSD 기반의 RAID-6 시스템과 하드디스크 기반의 RAID-6 시스템이 사용되었다.

표 2. 실험 환경  
Table 2. Experimental Environment.

항목	시스템 사양
CPU	Intel core 2 (2.4Ghz)
메모리	DDR2 1G * 2 개
운영체제	Linux kernel 2.6.24
Erasure 코드	Reed-Solomon, EVENODD, Liberation
SSD	HANA SSD(16 GB) * 6 개
HDD	Barracuda 7200.12 * 6 개

#### 2. Erasure codes 의 비트 갱신 횟수 측정

SSD 마모도 실험을 위해 1개의 20MB 텍스트 파일을 실험 데이터로 사용하여 비트 갱신 횟수를 측정하였다. 측정은 원본 파일과 Erasure 코드별로 인코딩된 데이터를 비트 단위로 비교 측정 하였다.

표 3은 SSD의 마모도를 측정하기 위해서 패리티 디스크의 비트 갱신 횟수를 비트 단위로 측정 결과이다. Liberation 코드를 적용하였을 경우 보다 EVENODD 와 Reed-Solomon 코드를 적용하였을 경우 비트 갱신 횟수가 각각 4.1%, 9.8% 많게 측정 되었다. 따라서 Liberation 코드가 SSD 환경에서 비트 갱신으로 인한 수명저하에 미치는 영향이 가장 적게 측정되었다.

표 3. 다양한 Erasure 코드들의 비트 갱신 횟수  
Table 3. bit update frequency for various of Erasure codes.

Erasure 코드	비트 갱신 횟수
Reed-Solomon 코드	11,174,520
EVENODD 코드	10,612,076
Liberation 코드	10,177,884

#### 3. 제안된 구조에 따른 인코딩 성능

제안된 Erasure 코드가 적용된 파일 시스템 레벨 구조와 디바이스 드라이버 레벨 구조의 전송 속도를 측정하기 위해서 200 MB 의 데이터 파일을 사용하여 인코딩 연산이 수행될 때의 성능을 측정하였다.

그림 9는 HDD 와 SSD 기반 RAID-6 시스템에 적용된 Erasure 코드들의 인코딩 성능을 보여준다.

우선 파일 시스템 레벨 구조의 성능을 비교해보면,

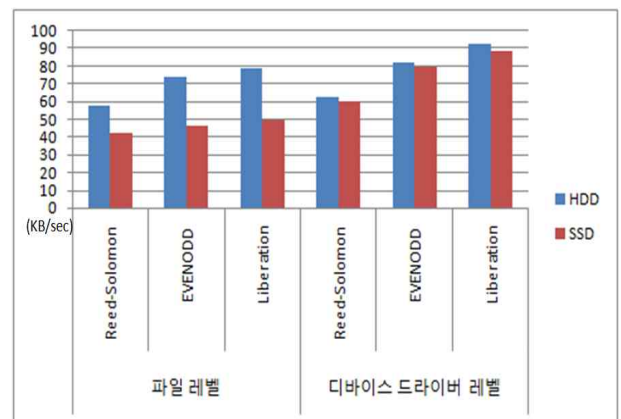


그림 9. Erasure 코드가 적용된 HDD 와 SSD 기반 RAID-6 시스템에서의 인코딩 성능  
Fig. 9. Encoding performance in HDD or SSD based RAID-6 system of Applying three erasure codes.

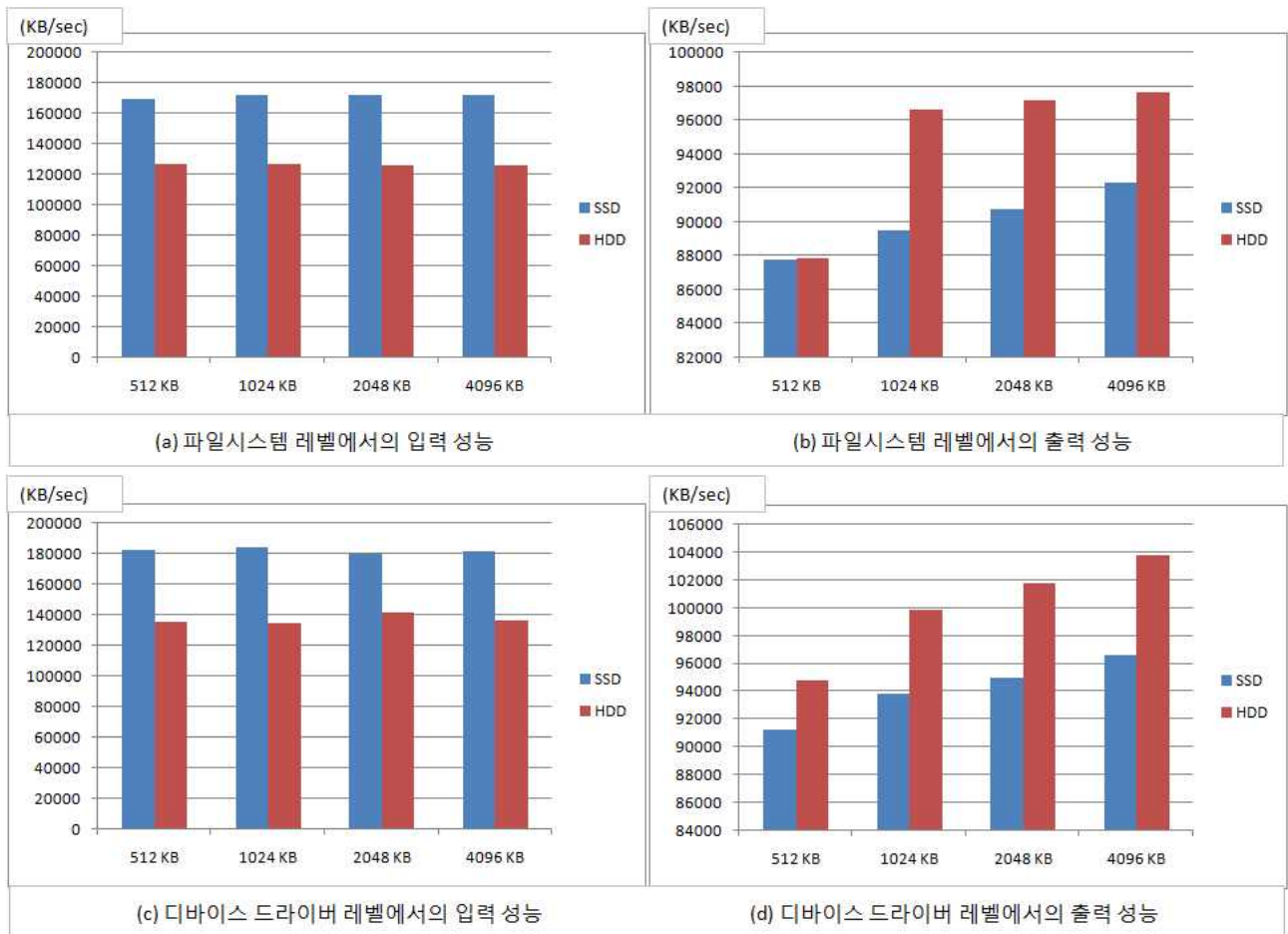


그림 10. 하드디스크와 SSD 기반에 적용한 RAID-6 시스템 구조의 I/O 성능  
 Fig. 10. I/O performance of Applied to HDD and SSD based RAID-6 architectures.

드디스크가 SSD 보다 평균 48.7 % 더 높은 인코딩 성능을 보여준다. 적용된 Erasure 코드들을 비교하였을 경우, SSD 기반에서는 Liberation 코드가 EVENODD 와 Reed-Solomon 코드들 보다 각각 5.1%, 8.9% 더 높은 성능을 보여주었고, 하드디스크 기반에서는 Liberation 코드가 EVENODD 와 Reed-Solomon 코드들 보다 각각 6.2%, 34.2% 더 높은 성능을 보여주었다.

그리고 디바이스 드라이버 레벨 구조의 인코딩 성능을 비교해 보면, 전체적으로 SSD 기반 보다 하드디스크 기반에 적용하였을 경우 평균 4.7% 더 높은 인코딩 성능을 보여준다.

적용된 Erasure 코드들을 비교하였을 경우, 하드디스크 기반에서는 Liberation 코드가 EVENODD 와 Reed-Solomon 코드들 보다 각각 5.8%, 37.4% 더 높은 성능을 보여준다. 그리고 SSD 기반에서도 Liberation 코드가 EVENODD 와 Reed-Solomon 코드들 보다 각각 6.4%, 39.1% 더 높은 성능을 보여주었다.

또한, 파일 시스템 레벨 보다 디바이스 드라이버 레벨에서 하드디스크는 평균 18.5% 향상된 전송 속도 성능을 보여 주었고, SSD 는 평균 57.1% 향상된 전송속도 성능을 보여주었다.

#### 4. 스트라이핑 모델에 따른 I/O 성능

그림 10은 제안된 두 가지 구조에 따른 입출력 성능을 나타낸다. 그림 10(a)와 그림 10(b)는 파일시스템 레벨에서의 입출력 성능을 보여준다. 그림 10(a)를 살펴보면 SSD 의 입력 성능이 하드디스크보다 평균 35% 더 빠른 성능을 보여준다. 출력성능은 HDD가 SSD 보다 평균 5% 더 빠르게 측정 되었다.

그리고 그림 10(c)와 그림 10(d)는 디바이스 드라이버 레벨에서의 입출력 성능을 보여준다. 그림 10(c)를 살펴보면 SSD의 입력 성능이 하드디스크보다 평균 32% 더 빠른 성능을 보여준다. 출력 성능은 HDD 가 SSD 보다 평균 6% 더 빠르게 측정 되었다.



디바이스 드라이버 레벨에 적용하였을 경우보다 파일시스템 레벨 구조에 적용하였을 경우, 입력성능은 SSD가 평균 6.4% 하드디스크가 평균 8.5% 향상 되었으며, 출력성능은 SSD가 평균 4.5% 하드디스크가 평균 5.5% 향상된 성능을 보여주었다.

5. Erasure 코드들에 따른 전력 소모

제안된 두 가지 구조에 Erasure 코드들을 적용하고 200MB의 데이터를 인코딩 할 때의 전력 소모를 측정하였다. 또한 측정 시뮬레이터를 사용하여 리눅스 기반의 SSD 전력 소모 측정 시뮬레이터<sup>[9]</sup>와 하드디스크 전력 소모 측정 시뮬레이터<sup>[10]</sup>를 사용하였다.

5.1. 파일 시스템 레벨 구조의 전력소모

그림 11은 하드디스크 기반과 SSD 기반에서 다양한 Erasure 코드를 파일 시스템 레벨에 적용 하였을 때, 전력 소모를 보여준다. 평균적으로 SSD가 하드디스크보다 44.3% 절감된 전력 소모 성능을 보여주었다.

적용된 Erasure 코드별 전력 소모를 비교하였을 경우, 하드디스크에서는 Liberation 코드가 EVENODD 와 Reed-Solomon 코드들 보다 각각 8.3%, 31.5% 적은 전력 소모를 보여준다. SSD 에서도 Liberation 코드가 EVENODD 와 Reed-Solomon 코드들 보다 각각 8.8%, 31.2% 적은 전력 소모를 보여 주었다.

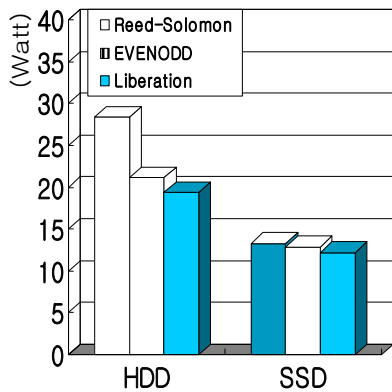


그림 11. 파일 시스템 레벨 구조의 전력소모  
Fig. 11. Energy consumption in File System Level.

5.2. 디바이스 드라이버 레벨 구조의 전력소모

그림 12는 하드디스크 기반과 SSD 기반에서의 다양한 Erasure 코드를 적용 하였을 때, 전력 소모를 보여 준다. 평균적으로 하드디스크보다 SSD 가 64% 더 낮

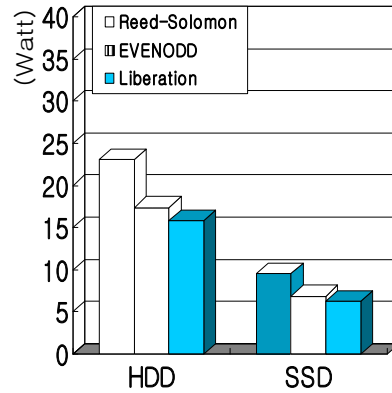


그림 12. 디바이스 드라이버 레벨 구조의 전력소모  
Fig. 12. Energy consumption in Device Driver Level.

은 전력 소모 성능을 보여준다.

적용된 Erasure 코드별 전력 소모를 비교하였을 경우, 하드디스크에서는 Liberation 코드가 EVENODD 와 Reed-Solomon 코드들 보다 각각 8.8%, 31.2% 낮은 전력소모를 보여준다. SSD 에서도 Liberation 코드가 EVENODD 와 Reed-Solomon 코드들 보다 각각 8.7%, 33.7% 낮은 전력 소모를 보여주었다.

또한, 파일 시스템 레벨 구조 보다 디바이스 드라이버 레벨 구조에서 하드디스크는 평균 18.5% 절감된 전력 소모 성능을 보여주고, SSD 는 평균 40.9% 절감된 전력 소모 성능을 보여주었다.

VI. 결 론

본 논문에서는 기존에 하드디스크 기반의 RAID 시스템에서 사용되는 Erasure 코드를 SSD 기반 RAID 시스템에 적용하는 두 가지 구조를 제안한다.

제안된 두 가지 SSD 기반 RAID-6 구조에 대한 성능 평가에서 파일 시스템 레벨 구조에 적용하였을 경우보다 디바이스 드라이버 레벨 구조에 적용하였을 경우가 더 좋은 성능을 보여 주었다.

또한 SSD의 수명제한 특성을 고려하여 최적화된 Erasure 코드를 찾기 위해 비트 갱신 횟수를 측정 하였다. 측정한 결과로 Liberation 코드를 적용하였을 때 비트 갱신 횟수가 가장 낮게 측정되었다. 그리고 인코딩 성능 및 전력소모를 비교해 보았을 때, Reed-Solomon 코드와 EVENODD 코드를 적용하였을 때 보다 Liberation 코드의 인코딩 성능이 더 좋은 성능을 가졌으며, 빠른 성능으로 인해 전력소모가 절감되었다.

따라서 Liberation 코드가 다른 기법들에 비해 SSD 기반의 스토리지 시스템에 적합한 Erasure 코드임을 알 수 있었다.

### 참 고 문 헌

- [1] Katz, R., Gibson, G., and Patterson, D., "Disk System Architectures for High Performance Computing." Technical Report, University of California, Berkeley, pp. 1-39, March 1989.
- [2] J. S. Plank. "A tutorial on Reed-Solomon coding for fault-tolerance in RAID-like systems.", Software - Practice & Experience, pp, 995 - 1012, September 1997.
- [3] M. Blaum, J. Brady, J. Bruck, and J. Menon. "EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures". In Proc. of the Annual International Symposium on Computer Architecture, pp. 245-254, 1994.
- [4] James S. Plank, "The RAID-6 liberation codes", Proc. of the 6th USENIX Conference on File and Storage Technologies, San Jose, California, pp.1-14, February 2008.
- [5] Kwanghee Park, Dong-Hwan Lee, Youngjoo Woo, Geunhyung Lee, Ju-Hong Lee, Deok-Hwan Kim, "Reliability and performance enhancement technique for SSD array storage system using RAID mechanism", Communications and Information Technology, 9th International Symposium, Incheon, Korea, pp. 140-145, January 2010.
- [6] D.A. Patterson, G. Gibson, and R.H. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)," Proc. ACM SIGMOD Conf., pp. 109 - 116, 1988.
- [7] R. H. Katz, G. A. Gibson, and D. A. Patterson, "Disk System Architectures for High Performance Computing," EECS Department, University of California, Berkeley, Tech. Rep. pp. 89-497, March 1989..
- [8] Asim Kadav, Mahesh Balakrishnan, Vijayan Prabhakaran, Dahlia Malkhi, "Differential RAID: Rethinking RAID for SSD Reliability", Workshop on Hot Topics in Storage and File Systems (HotStorage' 09) colocated with SOSP, October 2009.
- [9] 송재석, 양준식, 고영욱, 조원희, 김덕환, "IPTV 기반 STB를 위한 HDD와 SSD 저장장치의 성능평가", 대한전자공학회 2009년 하계종합학술대회, 605쪽~606쪽, 2009년. 7월.
- [10] 고영욱, 이근형, 박광희, 김덕환, "모바일 컴퓨팅 시스템에서 하드디스크의 전력 소모 측정 시뮬레이터 설계 및 구현", 한국정보과학회 학술발표논문집 2008 가을 학술발표논문집, 제35권 제2호 (B), pp. 459-463, 2008. 10.

### 저 자 소 개



송 재 석(학생회원)  
2009년 대전대학교 전자공학과  
학사 졸업.  
2009년~현재 인하대학교  
전자공학과 석사과정  
<주관심분야 : 임베디드 시스템,  
스토리지 시스템>



허 준 무(학생회원)  
2008년 인하대학교 전자공학과  
학사 졸업.  
2010년~현재 North Carolina  
State University  
Electrical Engineering -  
Master of Science 과정  
<주관심분야: 임베디드 시스템>



양 유 석(학생회원)  
2008년 순천향대학교 정보기술  
공학부 학사 졸업.  
2010년~현재 인하대학교  
전자공학과 석사과정  
<주관심분야 : 임베디드 시스템,  
스토리지 시스템>



김 덕 환(정회원)-교신저자  
2003년 한국과학기술원 컴퓨터  
공학 박사.  
2006년~현재 인하대학교  
전자공학부 부교수  
<주관심분야 : 시각정보처리, 스  
토리지 시스템, 임베디드 시스템>