韓國컴퓨터情報學會 論文誌
第16卷 第6號, 2011. 6.

2011-16-6-1-3

G-RAID: 대용량 저장장치에서 에너지 효율향상을 위한 그린 RAID 기법

김 영 환*, 석 진 선 **, 박 창 원**, 홍지만*

G-RAID: A Green RAID Mechanism for enhancing Energy-Efficiency in Massive Storage System

Young Hwan Kim*, Jin Sun Suck **, Chang Won Park **, Ji Man Hong*

요 약

각종 환경 규제에 대한 대응이 세계 IT 시장에서 큰 이수로 부각되면서 데이터 센터 유지를 위해 소모되는 막대한 양의 에너지와 저장 장치를 효율적으로 관리하기 위한 통합(Consolidation), 가상화(Virtualization), 최적화(Optimization), 관리 (Leverage) 등의 다양한 시도가 이루어지고 있다. 본 논문에서는 이러한 노력의 일환으로 다수의 저장 장치를 계층화하여 전력 소비량 및 데이터 이중화 작업의 강도를 달리하는 새로운 개념의 RAID 기법을 제안한다. G-RAID는 저장 장치의 전력 소비량을 계층화하고 데이터 접근 빈도수에 따라 저장 장치를 선택함으로써 데이터 I/O 성능 저하의 발생을 최소화하면서 전력 소비를 최소화 했다. 또한 파일 시스템과 G-RAID의 블록 맵 정보는 효율적인 작업을 위해 연속적으로 위치한 블록들의 그룹을 기본 단위로 처리되며 그룹 별 접근 빈도수를 고려하여 블록 연결 작업으로 인한 성능 감소를 최소화하고자 했다. 실험 결과, 블록 재배치 작업으로 인한 응답 시간의 향상이 나타났지만 전력 소비 율은 최대 38% 감소하는 것으로 나타났다.

▸ Keyword : 저장장치, 계층화, 전력소비량, 복수배열독립디스크, 블록 재배치

Abstract

In the global IT market, a lot of issues for responding to various environmental regulations emerged. In case of the data centers, it is consuming huge amounts of energy to maintain. So there have been various technical attempts as Consolidation, Virtualization, Optimization to efficiently

• 제1저자 : 김영환 • 교신저자 : 석진선, 박창원, 홍지만
• 투고일 : 2011. 01. 31, 심사일 : 2011. 02. 09, 게재확정일 : 2011. 02. 14
* 숭실대학교 컴퓨터학과(Dept. of Computing, Soongsil University)
** 전자부품연구원 융합신산업연구본부(Convergence Emerging Industries R&D Division, Korea Electronics Technology Institute)
※ This study was supported by "NGS(Next Generation Storage) System R&D" funded by the Ministry of Knowledge Economy

manage energy and data storage to fix the problems. In this paper, we propose a new RAID(Redundant Array of Independent Disks) mechanism which is differing the intensity of power consumption and works to provide data protection and disaster recovery(backup, mirroring etc.) to stratify multiple volumes. G-RAID minimize the power consumption and the lower of I/O performance by selecting the volume depending on the frequency of data access while classifying the power consumption between volumes in storage system. Also, it is possible that a filesystem and block map information of G-RAID is processed by basic unit which is group located in a row for the blocks to work efficiently and can minimize the performance degradation of block mapping load by the access frequency in each groups. As a result, we obtained to elevate a little bit of response time caused by block relocation work, but showed the decrease of power consumption by 38%.

▸ Keyword : Storage System, Stratification, Power Consumption, RAID, Block Relocation

I. 서 론

최근 많은 나라들이 지구 온난화 문제로 인해 환경 규제 증가 및 에너지 효율성 제고에 관한 필요성을 인식하면서 친환경 IT의 중요성이 부각 되고 있다.

미국 내 데이터 센터에서의 2000~2006년까지 전력 사용량을 보면 저장장치와 볼륨 서버 분야가 데이터 센터 요소 중에서 연평균 증가율이 각각 20%, 17%로 가장 높음을 보여준다[1]. 가장 대표적인 에너지 소비 기기는 웹 서버와 관련 설비 및 네트워크 장비로 알려져 있으며 운영에 필요한 전력량 65%를 각 서버에서의 발열을 냉각하기 위한 냉방 시스템에서 소모하는 것으로 조사되었다.

이와 같은 이유로 최근 데이터 센터 내 서버, 스토리지 관련하여 에너지 효율 향상을 위한 기술이 많이 연구되고 있는데, 그 예로 Gear-shifting 방식을 이용하여 디스크의 전력 소비량을 줄이는 PARAID, 디스크의 idle 상태에서의 스핀-다운 관리를 통한 MAID, 램 버퍼와 디스크 버퍼를 이용하여 대형 병렬 저장장치 시스템의 에너지 효율 향상을 위한 Energy-Efficient Framework, 서버에서의 파일 접근 빈도가 상이함을 이용하여 접근빈도가 낮은 파일에 대해서는 저속의 디스크에서 관리하는 PDC와 같은 기술이 대표적이다.

본 논문에서는 앞서 설명한 기존 기술과는 달리 각 서버의 전력 소모량을 줄여 데이터 센터의 에너지 절약화를 위한 G-RAID를 제안한다. 기존의 방식들이 저장 장치에 대한 접근 빈도수를 이용하여 저장 장치의 전력량을 계층화한데 비해 G-RAID는 각 데이터의 접근 빈도수를 이용하여 저장 장치의 접근 빈도수를 직접 관리함으로써 계층화된 저장 장치를 보다 효율적으로 사용하여 에너지 절약 효과를 극대화하고자 했다. G-RAID는 I/O 성능 및 데이터 신뢰성(reliability)을 유지하면서 전력 소모량을 줄이기 위해 제안된 새로운 다중 저장 장치 관리 기법이다. 이는 다수의 볼륨을 계층화하고 각 계층마다 사용되는 전력량에 차등을 두어 전력 소비량을 감소시킨다. 예를 들어 N개의 계층으로 구성되는 경우, 각 계층의 저장 장치는 레벨이 낮을수록 스핀-다운(spin-down)으로 인한 전력 소비량이 감소하게 된다. 또한 스핀 다운으로 인한 I/O 성능 저하를 예방하기 위해 상대적으로 접근 빈도수가 높은 데이터를 상위 계층에 유지함으로써 성능을 유지하고 하위 계층의 스핀-다운 상태를 최대한 지속시켜 저장 장치의 상태 변화로 인한 전력 소모가 발생하지 않도록 한다. 각 계층 별 데이터 복구 기법은 저장된 데이터 블록에 대한 접근 빈도수를 고려하여 적합한 RAID 기법을 선택함으로써 디스크 효율성 및 데이터 신뢰성을 향상 시킨다.

본 논문의 2장에서는 저장장치 관련하여 에너지 관점에서 연구된 다양한 기존 기술에 관해 분석하고, 3장에서는 본 논문에서 제안한 저장장치의 에너지 효율 향상을 위한 Green RAID의 구조, 4장에서는 실제 구현한 G-RAID에 대한 성능 평가를 수행한 결과 그리고 5장에서는 결론과 함께 향후 작업으로 구성된다.

II. 관련 연구

친환경 IT를 향한 다양한 노력은 서버/스토리지에 대한 성능과 저장장치 운용관리의 효율화를 통한 서버의 에너지 절감이라는 두 가지 관점에서 진행되고 있다.

1. PARAID

PARAID [2]는 서버에서 사용되는 디스크의 전력 소비량을 줄이기 위해 제안된 새로운 방식의 RAID로 특화된 하드

웨어를 사용하지 않고 스트라이핑 패턴을 왜곡하는 Gear-shifting 방식을 사용하여 에너지 소모를 최소화한다.

Gear-shifting 방식은 사용되는 저장장치의 개수에 따라 gear-level을 달리하여 데이터 I/O 대역폭을 조절하는 방식으로 I/O 요청량에 따라 대역폭을 조절하여 저장장치를 idle한 상태로 유도하는 것이 가능하다. Gear변환은 upshift와 downshift 작업에 의해 수행되며 작업이 발생하는 기준이 되는 T는 일정시간 동안 측정한 시스템 부하 평균 값(U)과 편차 값(S)을 이용하여 결정 된다.

1.2 MAID

Massive Arrays of Idle Disks (MAID) [3] 는 각 저장장치의 상태 정보를 활용하여 전력 소모량을 조절하는 시스템이다. 저장장치가 일정 시간 이상 idle한 상태를 지속하는 경우, 저장장치의 전력 소비량을 줄이기 위한 스핀-다운을 시도하며 일단 스핀-다운된 저장장치의 상태를 최대한 지속시키기 위해 스핀-업 상태를 항상 유지하고 있는 특정 저장장치를 캐시로 활용했다. 이런 시도는 I/O 성능을 크게 저하시키지 않으면서 전력 소비량을 크게 감소시키는 성과를 보였다. 하지만 데이터의 지역성이 미약한 경우 캐시에서 실패(cache miss) 횟수가 증가하여 전력 소비량이 증가하는 한계를 극복하지 못했다.

1.3 Energy-Efficient Framework

대형 병렬 저장 시스템[4]의 에너지 효율을 향상시키기 위한 프레임워크로 디스크의 3가지 상태 (active, idle, sleep)가 가진 각자의 에너지 사용량을 기반으로 각 디스크 상태를 가능한 오래 유지하기 위해 활성화 된 디스크를 최대한 활용하며 이를 위해 저장 장치 외에 '램 버퍼'와 '디스크 버퍼'를 추가적으로 사용한다.

쓰기 요청 시, 데이터의 사이즈를 확인하여 대용량인 경우에 바로 저장 장치에 쓰기를 요청하고, 작을 경우 '램 버퍼'에 기록하며 일정 시간 뒤에 '디스크 버퍼'로 옮겨진다. 디스크 버퍼는 특정 영역을 각 저장 장치에 맵핑하여 쓰기 요청이 발생하는 경우에 저장 장치 대신 사용되며 이러한 방법은 요청이 발생한 저장 장치 외의 저장 장치들을 사용하지 않게 하여 에너지를 절약한다. 읽기 요청의 경우, 프레임워크는 램 버퍼, 디스크 버퍼, 저장 장치 순으로 읽기 요청을 하여 저장 장치의 사용을 가능한 줄인다. 이 프레임워크는 38%의 에너지를 절약하는 결과를 얻었다.

1.4 PDC

PDC [5]는 네트워크 서버의 파일들이 액세스 빈도가 고르지 않다는 점을 이용하여 에너지 효율을 높이는 기법이다. 일반적으로 몇몇의 파일만이 액세스 빈도가 높고 다수의 파일은 액세스 빈도가 매우 낮다는 점을 이용하여 액세스 빈도가 높은 파일들은 액세스가 높은 디스크로 이동시키고 그에 반해 액세스 빈도가 낮은 파일들만을 가진 디스크는 저 전력 모드로 사용한다.

NomadFS는 PDC를 구현한 것으로 다중 큐를 이용하여 파일들의 액세스 빈도를 관리하며 각 큐는 하나씩 디스크와 맵핑되어 큐에 있는 파일들을 해당하는 디스크로 이동시킨다. 가장 상위 큐는 가장 액세스 빈도가 높은 파일들만 위치하여 맵핑된 디스크를 계속하여 사용될 수 있고, 반면 하위 큐에 맵핑된 디스크는 사용 빈도가 줄어 스핀-다운되어 사용된다.

이 방식은 액세스 빈도가 낮은 파일을 가진 디스크의 소비 전력이 줄어들어 에너지 효율을 높일 수 있지만 몇몇의 디스크에만 과부하가 생길 수 있어서 그에 따른 성능 하락 가능성이 있다.

PARAID와 Energy-Efficient Framework는 접근 빈도수와 같은 데이터만의 특성을 전혀 고려하지 않기 때문에 각 데이터의 접근 빈도수에 따라 계층화된 저장 장치를 사용함으로써 전력 소비 감소율을 극대화하고 I/O 성능의 저하를 최소화하는 G-RAID와는 다른 방식이라고 할 수 있다. 반면 MAID는 저장 장치를 계층화하여 전력 소비를 차별화한다는 점은 G-RAID와 동일하지만 각 저장 장치에 위치할 데이터를 선별한다는 점에서 서로 다른 방식이라고 할 수 있다. PDC는 데이터 접근 빈도수를 고려하여 저장 장치의 전력 소비를 계층화하는 방식을 사용함으로써 G-RAID와 거의 유사한 방식을 제안했지만 파일 시스템 계층에서 제안한 방식으로 파일 시스템과의 비 종속성을 위해 블록 디바이스 계층에서 블록을 기본 단위로 처리하는 G-RAID와는 분명한 차이를 보인다.

III. Green RAID의 설계

G-RAID의 주요 설계 목적은 I/O 성능, 데이터 신뢰성 유지 그리고 에너지 절약화이다.

1. 에너지 절약화

슈퍼 컴퓨팅 환경에서 전체 데이터의 50%는 write 작업 이후에 절대 접근되지 않으며 전체 25%에 해당되는 데이터는 기록 후 단 한번만 접근된다[6]. 캐시(cache) 작업의 효율성을 뒷받침하는 근거로 자주 사용되는 분석 결과로 실제 빈번하게 접근되는 HOT 데이터는 극히 일부의 데이터임을

의미한다. G-RAID는 이러한 가정을 기반으로 소량의 HOT 데이터를 스핀-업(spin-up) 상태를 유지하고 있는 상위 계층의 저장 장치에 저장하고 대 다수의 COLD 데이터를 스핀-다운 상태를 유지하고 있는 하위 계층의 저장 장치에 저장함으로써 전력 소모량을 최소화하고자 했다.

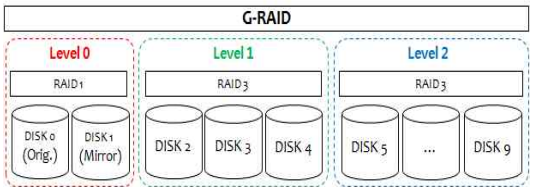


그림 1. 에너지 절약화를 위한 G-RAID의 계층화 전략
Fig. 1. G-RAID structure for saving power consumption

그림 1은 에너지 절약을 위해 다수의 볼륨을 계층화하여 관리하는 G-RAID의 구성 예제이다. 총 10개의 디스크를 3개의 계층으로 나누어 관리하는 형태로 'DISK0'과 'DISK1'이 하나의 볼륨으로 묶여 Level0으로 구성되었다. Level0은 최상위 계층으로 접근 빈도수가 가장 높은 데이터만을 저장하며 저장 장치를 스핀-업 상태로 유지하기 때문에 가장 많은 전력을 소비하는 계층이다. Level1은 'DISK2'-'DISK4'로 구성되며 접근 빈도수가 중간 정도인 WARM 데이터를 저장한다. 'DISK5'-'DISK9'는 볼륨으로 묶여 Level2로 설정된다. Level2는 접근 빈도수가 거의 발생하지 않는 COLD 데이터만을 저장 관리하며 디스크를 스핀-다운 된 상태로 유지하기 때문에 G-RAID 계층 중 가장 적은 에너지를 소비한다.

2. 데이터 신뢰도

데이터 손실은 데이터를 기록하는 과정에서 주로 발생하게 된다. 따라서 HOT 데이터를 다량 보유하고 있는 상위 계층에 위치하는 저장 장치의 데이터 손실의 가능성은 하위 계층에 위치하는 저장 장치보다 높다.

G-RAID는 이러한 가정을 기반으로 상위 계층에 위치하는 볼륨의 관리 기법으로 다수의 추가적인 저장 장치를 필요로 하지만 손실된 데이터를 완벽하게 복구하는 방식을 사용하고 하위 계층에 위치하는 볼륨에는 데이터 복구 능력이 상대적으로 미약하지만 상대적으로 적은 개수의 추가 저장 장치를 필요로 하는 기법을 사용하여 저장 장치를 최소한으로 사용하면서 데이터 신뢰성을 유지하고자 했다. 따라서 G-RAID의 구성 예인 그림 1은 Level0에서는 RAID-1을 사용했으며 나머지 하위 계층에서는 RAID-3을 사용하였다.

표 1. 각 RAID의 특성
Table 1. RAID Specification

	# of check disks	Reliability
RAID-1	N	1
RAID-2	C>= $\log_2(N\_G+C+1)$	$R_{level}(t)$ ($\alpha = 0$)
RAID-3,4,5	1	$R_{level}(t)$

표 1은 각 RAID를 구성하기 위해 필요한 추가 저장장치 및 데이터 신뢰도를 정리한 것이다. N은 원본시스템을 구성하는 디스크의 개수를 의미하는 것으로 RAID-1을 구성하기 위해서는 원본시스템을 구성하고 있는 디스크만큼의 추가 디스크가 필요함을 알 수 있다. 반면 Hamming 오류 정정 코드를 사용하여 데이터 신뢰성을 관리하는 RAID-2는 각 디스크 그룹(G)마다 원본데이터를 저장하는 디스크(N_G)와는 별도로 C >= log2(N_G+C+1) 수식을 만족하는 C개의 여분 디스크를 필요로 한다. RAID-3,4,5는 패리티를 구성하여 데이터 신뢰성을 유지하는 기법으로 디스크 그룹 당 1개의 여분 디스크를 필요로 한다.

식(1)은 [7]에서 인용한 것으로 laplace transform approach를 사용하는 경우의 신뢰도를 의미하며 이를 이용하여 G-RAID의 볼륨 신뢰도를 식(2)와 같이 표현했다.

G-RAID를 구성하는 3-계층은 서로 다른 접근 빈도수를 가지기 때문에 서로 다른 fail 발생률 (F_{level})을 가지며 이에 따라 적절한 RAID 방식을 선택함으로써 데이터 신뢰성 저하를 예방한다. 각 계층의 데이터 신뢰성(R_{level})은 식(2)의 (a), (b)와 같은 특성을 가지며 F_{level}은 각 계층 별 접근 빈도수 차이에 따라 식(2)의 (c), (d)와 같은 특성을 가진다.

$$R_{level}(t) = A_1 e^{\beta_1 t} + A_2 e^{\beta_2 t}$$

$$\beta_1 = \frac{-(((N+C)(2-p\alpha)-1)\lambda+\mu)+\sqrt{X}}{2}$$

$$\beta_2 = \frac{-(((N+C)(2-p\alpha)-1)\lambda+\mu)-\sqrt{X}}{2}$$

$$A_1 = \frac{\beta_1+\mu+(N+C-1)\lambda}{\beta_1-\beta_2} \quad (1)$$

$$A_2 = \frac{\beta_2+\mu+(N+C-1)\lambda}{\beta_2-\beta_1}$$

$$X = ((N+C)((N+C)\rho^2\alpha^2-2\rho\alpha)+1)\lambda^2+\mu^2 + 2((N+C)\rho(2-\alpha)-1)\lambda\mu$$

$$R_{GRAID}(t) = \prod_{level=1}^{L} \{1-(F_{level}\times(1-R_{level}))\} \quad (2)$$

$$0 < R_{level} \le 1 \quad (a)$$

$$\lim_{level\to 0} R_{level} = 1 \ , \ \lim_{level\to\infty} R_{level} = 0 \quad (b)$$

$$0 \le F_{level} \le 1 \quad (c)$$

$$F_0 > F_1 > \ldots > F_{L-1} \quad (d)$$

3. 성능유지

G-RAID는 접근 빈도수에 따라 각 데이터 블록을 분류하여 적합한 계층의 저장 장치에 기록한다. 따라서 파일 시스템의 블록 할당 정책에 의해 할당된 논리적 블록 번호와 실제 데이터가 기록되는 물리적 블록 번호는 서로 다르며 이러한 문제점을 해결하기 위해 재 연결 작업이 반드시 필요하다. 하지만 재 연결 작업은 모든 I/O 작업에 추가적인 부하를 발생시켜 성능 저하를 유발 할 수 있고 파일 시스템이 디스크 접근 시간을 최소화하기 위해 연속적으로 배치한 블록의 위치를 이동시킴으로써 대기 시간을 유발 할 수 있다.

이러한 한계를 극복하기 위해 G-RAID는 블록들을 그룹 단위로 묶어서 처리함으로써 재 연결 작업으로 인한 부하를 최소화하고 파일 시스템에 의한 블록 연속성을 최대한 유지하고자 노력했다.

모든 계층을 구성하는 저장 장치들은 총 G_n개의 그룹으로 나뉘어 관리되며 각 그룹 크기는 G_s로 모두 일정하다. 각 그룹은 연속적으로 배치된 G_c개의 블록으로 구성되며 그룹별 접근 빈도수(G_a)와 빈도수 분포도(G_d) 값을 가진다. 접근 빈도수는 해당 그룹에 발생한 총 접근 수를 의미하며 빈도수 분포도는 실제 접근된 블록 수(N_a)의 비율을 의미하는 것으로 식(3)과 같이 표현 가능하다. N_a가 '1'에 가까울수록 그룹 내의 모든 블록에 대한 접근이 일어남을 의미하고 '0'에 가까울수록 그룹 내의 특정 블록에 접근 빈도수가 집중됨을 의미한다.

$$G\_d = \frac{N\_a}{G\_c} = \frac{N\_a}{\frac{G\_s}{Block\_Size}} \quad (3)$$

G_a 가 기준치를 넘고 G_d가 '1'에 가까운 경우 그룹 안에 속한 모든 블록에 대한 접근 빈도수가 높은 것임으로 그룹이 HOT 영역으로 이동시킨다. 그룹 단위로 이동하기 때문에 데이터 연속성을 유지한 상태로 '블록 계층화'가 가능하다.

4. Green RAID의 구성요소

G-RAID는 블록 상태 관리자, 계층 관리자, 그리고 맵핑 관리자의 3가지 구성 요소를 가진다. '블록 상태 관리자' 와 '계층 관리자'는 각각 접근 빈도수를 기준 값으로 각 데이터 블록의 상태를 결정하고 적절한 저장 위치 선정 및 데이터 블록의 저장 위치 변경 작업을 수행한다. '맵핑 관리자'는 G-RAID에 의해 선정된 물리적 블록 번호를 파일 시스템의 논리적 블록 번호와 연결하여 블록의 계층화 관리가 파일 시스템과 독립적으로 동작할 수 있도록 한다.

4.1 블록 계층화 관리

각 데이터는 접근 빈도수에 따라 3가지 상태(HOT,

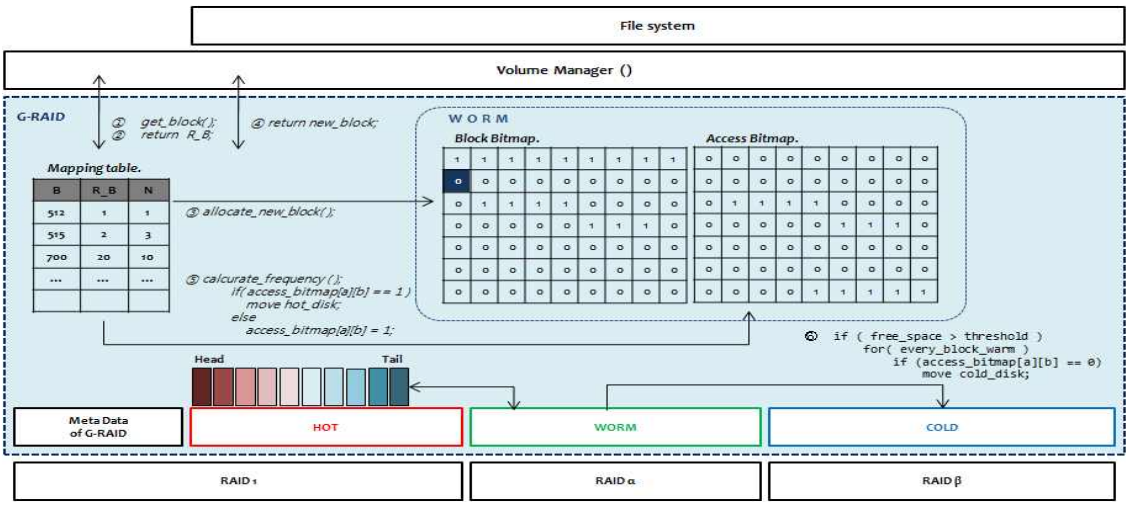


그림 3. G-RAID의 세부 구성도
Fig. 2. G-RAID Components

WARM, COLD)로 분류되며 총 3개의 계층으로 이루어진 저장 장치에 나뉘어 저장된다. 각 저장 장치는 계층마다 서로 다른 RAID 정책을 사용함으로써 데이터 신뢰성을 유지하면서 공간 활용도를 높여 전력 소비량을 최소화한다.

모든 쓰기 작업은 WARM 저장 장치에서 수행되며 각 데이터 블록은 일정 기간 후에 접근 빈도수에 의해 저장 위치를 재결정한다. 이는 전체 데이터의 50%는 생성 후 접근되지 않는 COLD 데이터라는 가정에 기반을 둔 것으로 데이터의 이동 작업을 최소화하고 스핀-다운된 저장장치의 상태를 최대한 유지하여 전력 소비를 감소시키기 위한 방식이다.

$$PES = 1 - \frac{\sum_{level=0}^{N} (V_{level} \times P_{level})}{DP_0} \quad (4)$$

식(4)는 G-RAID의 계층화 구조를 사용함으로써 발생하는 전력 소비 감소율을 표현한 것이다. 'N'은 G-RAID가 볼륨을 에너지 절감 성능에 따라 계층적으로 관리하기 위해 부여한 전체 계층을 의미하는 것으로 본 논문에서는 RAID 1, RAID α, RAID β의 3계층으로 분류하고 있으며 'V_{level}'은 각 계층의 볼륨을 구성하는 디스크의 개수 그리고 'P_{level}'은 각 계층의 볼륨이 사용하는 전력 소모량을 의미한다. 'D'는 전체 디스크 개수를 의미하는 것으로 $D = \sum_{i=0}^{N} V_i$ 과 같이 정의할 수 있다. 계층은 증가할수록 전력 소모량이 감소하며 가장 낮은 계층의 전력 소모량인 P_0은 디스크가 정상적으로 동작하는 경우 소모되는 최대 전력량을 의미한다. 각 계층의 전력소비량 간에는 $P_0 > P_1 > ... > P_{n-1} > P_n$와 같은 관계가 성립되며 전 계층의 저장장치가 모두 정상 동작하는 경우 최대 전력 소비량은 $DP_0 = \sum_{i=0}^{N} V_i P_0 = 1$ 과 같다.

4.2 블록 재연결

재연결 작업은 접근 빈도수에 의해 변경된 각 블록의 저장위치를 파일시스템과 공유하기 위한 것으로 파일시스템이 인지하는 논리 블록번호와 실제 저장장치의 물리적 블록번호를 연결하기 위한 작업으로 블록 계층화 관리를 위한 블록 재배치 작업이 발생하는 경우에도 파일시스템과의 연동을 위해 필요하다.

재연결 작업을 위한 그림 2의 연결 테이블(Mapping table)은 3가지 요소인 논리 블록 번호(B)와 물리적 블록 번호(R_B) 그리고 연속된 블록의 개수(N)로 구성된다. 예를 들어 (B, R_B, N)이 (515, 2, 3)인 경우, 연속된 3개의 블록들이 논리 블록 번호 515와 물리 블록 번호 2를 시작점으로 하여 515->2, 516->3, 517->4와 같이 차례로 할당 된다. 물리적 블록은 '블록 비트맵(Block Bitmap)'을 이용하여 할당된다. 각 비트는 하나의 블록을 의미하며 '0'은 블록이 비어 있음을 의미하고 '1'은 블록이 사용 중임을 의미한다.

4.3 접근 빈도수 측정

접근 빈도수 측정 작업은 WARM 저장 장치에서만 수행되며 이는 "전체 파일의 50%에 해당하는 파일이 생성 후 전혀 접근되지 않는다."는 가정 하에 COLD 저장 장치의 상태를 가능한 오래 유지하여 전력 소비를 최소화하고 HOT 저장 장치의 여유 공간을 확보하기 위한 재배치 작업 횟수를 최소화하여 I/O 성능 저하를 예방하기 위해 선택한 정책이다.

접근 빈도수는 그림 2의 '접근 비트맵(Access Bitmap)'에 의해 관리된다. 각 비트는 WARM 저장 장치에 위치하는 블록을 의미하며 데이터 접근이 발생하는 경우에 해당 블록의 비트 값을 '1'로 변경하여 접근이 있었음을 기록한다.

4.4 블록 재배치

재배치 작업은 각 데이터 블록의 상태(HOT/WARM/COLD)에 따라 저장 위치를 변경하는 작업으로 적합한 데이터 복구 정책을 선택함으로써 데이터 저장 공간을 효율적으로 사용하고 이로써 데이터 신뢰성을 유지하면서 전력 소비를 줄이는 G-RAID의 핵심 작업이다.

하지만 빈번한 데이터 이동 작업은 전체 시스템의 I/O 성능 저하 및 전력 소비를 증가시킬 수 있기에 다음과 같은 규칙에 의해 동작한다. HOT 저장 장치는 LRU 알고리즘을 이용하여 데이터 블록의 대략적인 접근 빈도수를 측정하며 이를 기준으로 HOT 저장 장치 이용률이 기준 값 $\alpha\%$ 이상이 되는 경우에 WARM 저장 장치로 해당 데이터 블록을 이동시킨다. WARM 저장 장치는 '접근 비트맵'에 의해서 접근 빈도수를 측정하며 저장 장치의 이용률이 기준 값 $\beta\%$ 이상이 되는 시점에 현재 WARM 저장 장치에 위치하면서 접근 비트 값으로 "0" 값을 가지는 모든 블록을 COLD 저장 장치로 이동시킨다. 반면에 요청된 데이터 블록의 접근 비트 값이 현재 "1"로 설정되어 있는 경우, 해당 데이터 블록을 HOT 저장 장치로 바로 이동시킨다. 이는 접근 율이 높은 HOT 데이터를 스핀-업 상태의 저장 장치에 유지함으로써 성능 및 전력 부분에 대한 이득이 데이터 이동 작업으로 인해 발생하는 손실보다 클 것이라는 예측에 의해 결정된 사항이다. COLD 저장 장치는 WARM 저장 장치에서 일정 시간 동안 접근이 없었던 데이트 블록만이 저장 된 영역인 만큼 접근 발생률이 현저히 낮을 것으로 예상된다. 하지만 접근이 발생하는 경우 해당 데이터 블록은 바로 WARM 저장 장치로 이동되어 후속 처리에 대비한다.

IV. 구현 및 성능평가

G-RAID 실험 환경은 disksim [8]을 이용하여 그림 3과 같이 구성되었다.

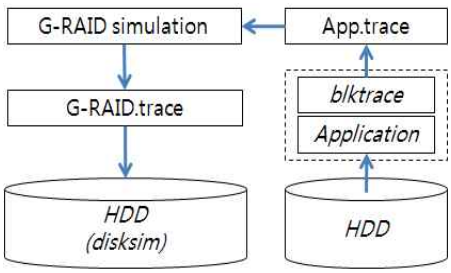


그림 4. G-RAID 실험 구성도
Fig. 3. G-RAID Test Diagram

G-RAID 시뮬레이션은 HOT 데이터와 WARM 데이터를 위해 각각 “블록 크기 * 100”, “블록 크기 * 200”의 공간을 할당했으며 COLD 데이터를 위한 저장 공간은 여유롭게 할당하여 여유 공간 부족으로 인한 작업 실패가 발생하지 않도록 했다. 그림 3에서 나타낸 것과 같이 blktrace [9]를 이용하여 얻은 App.trace 파일에 ‘블록 계층화 관리’ 및 ‘블록 재연결’ 등의 G-RAID의 특성화 작업 수행을 반영하여 G-RAID.trace 파일을 생성한다.

실험에서 사용된 IOzone [10]은 블록 사이즈8Kbytes로 고정하고 파일 사이즈를 512Kbytes, 1Mbytes, 2Mbytes, 4Mbytes, 8Mbytes 그리고 16Mbytes로 변화시키면서 write, re-write, read, re-read, random-write 그리고 random-read 작업을 수행하였다. PostMark[11]는 오래도록 보관되지 않고 용량이 작은 파일들을 사용하는 인터넷 소프트웨어에서 사용되는 파일들에 대한 성능 측정을 위한 합성 마이크로 벤치마크로 주어진 범위 내의 크기를 가지는 다량의 텍스트 파일들을 생성한 후, 랜덤하게 파일을 골라 삭제, 읽기, 추가(append) 등의 작업을 수행한다.

```
//WRITE

graid_set_bitmap( warm_bitmap );
graid_clear_bitmap( warm_cnt_bitmap );

//READ

if( input_blkno < GRAID_HOT )
        graid_LRU_move( );
 else if( input_blkno < GRAID_HOT + GRAID_WARM )
        if( graid_check_bitmap( warm_cnt_bitmap, input_blkno ))
                graid_set_bitmap( warm_cnt_bitmap, input_blkno );
else
        if( graid_check_free( hot_bitmap ))
                warm_to_hot( input_blkno );
        else if( (graid_check_free( warm_bitmap ) )
        {
                hot_to_warm( victim_of_hot( ) );
                warm_to_hot( input_blkno );
        }
        else
        {
                warm_to_cold( victim_of_warm( ) );
                hot_to_warm( victim_of_hot( ) );
                warm_to_hot( input_blkno );
        }
}
```

그림 5. G-RAID 의사 코드
Fig. 4. G-RAID Pseudo Code

1. 계층화 관리

그림 5은 IOzone의 워크 로드를 분석한 것으로 G-RAID의 설계 가정에 따라 최초의 블록 기록을 제외한 접근 횟수가 2번 이상인 데이터 블록을 HOT, 접근 횟수가 1번인 데이터 블록을 WARM 그리고 생성 후 전혀 접근 되지 않은 블록의 상태를 COLD라고 정의했다. Y축은 총 접근 블록 중 각 상태에 속하는 블록의 비율을 의미하며 X축은 각 데이터의 상태를 의미한다.

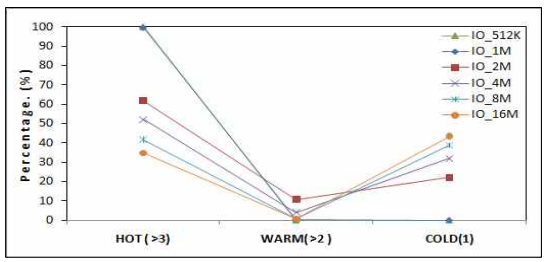


그림 6. IOzone 워크로드
Fig. 5. IOzone Workload

그림 6은 그림 5의 IOzone을 G-RAID에 적용한 경우 발생하는 워크로드로 3-계층으로 구성된 HOT, WARM, COLD 저장 계층에 발생하는 접근 비율을 표현한 것이다. 표 4의 결과와 유사하게 최대 파일 사이즈가 작을 수 록 HOT 영역과 WARM 영역의 접근 빈도수가 유사해지는 것을 볼 수 있으며 대체적으로 HOT 영역에 대한 접근 빈도수가 높은 것

을 알 수 있다. 이는 각 블록의 상태에 따라 저장 위치를 결정하는 G-RAID의 '블록 계층화 작업'이 효과적으로 동작하고 있음을 의미한다. 표 3보다 WARM 영역에 대한 접근 빈도수가 상대적으로 높은 것은 모든 데이터의 최초 쓰기 작업이 WARM 영역에서 처리되는 G-RAID의 정책과 HOT 영역의 제한에 따른 이동 작업(hot->warm)으로 인한 결과인 것으로 분석된다.

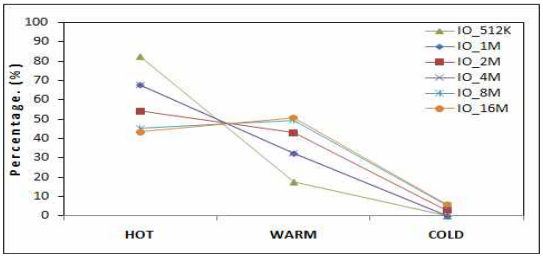


그림 7. IOZone를 적용한 G-RAID 워크로드
Fig. 6. G-RAID Workload with IOZone

그림 7은 PostMark의 워크로드를 표현한 것으로 단 기간 파일 사용이 두드러지는 인터넷 소프트웨어의 특성이 나타난다. 그림 8은 그림 7의 워크로드를 G-RAID에 반영한 결과 그래프로 일반적으로 파일 사이즈가 1Mbytes 이하인 경우에 WARM 영역에서 높은 빈도수를 보이고 HOT과 COLD 영역에서 낮은 빈도수를 보였으며 파일 사이즈가 증가하면서 COLD 영역에 대한 접근 빈도수가 증가하는 것으로 나타났다. 이러한 현상은 파일 사이즈가 증가하면서 구성 블록의 개수가 WARM 영역에 유지될 수 있는 한계를 벗어나 COLD 영역으로 이동했고 이러한 '블록 계층화 작업'이 이점으로 작용하기 위한 충분한 I/O가 발생하지 않아 나타난 현상으로 보인다. 충분한 I/O가 발생한다면 '블록 계층화 작업' 으로 인한 접근 빈도수는 상대적으로 낮은 비율에 머물게 될 것이다.

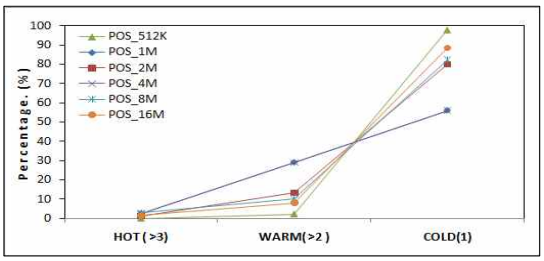


그림 8. PostMark 워크로드
Fig. 7. PostMark Workload

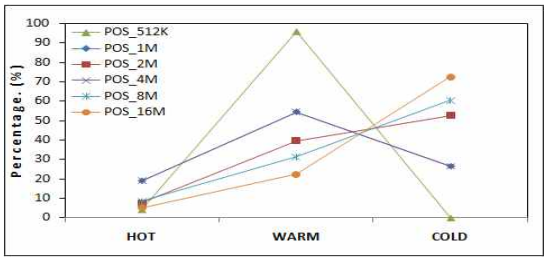


그림 9. PostMark를 적용한 G-RAID 워크로드
Fig. 8. G-RAID Workload with PostMark

2. 전력 소비량

표 2는 전력 소비량을 예측하기 위해 필요한 디스크 정보를 나타낸 것으로 PB-LRU [12]에서 사용한 값을 참조하였다. 전력량 예측을 위해 사용된 값은 'Active Power(Read/Write)', 'Idle Power@15000RPM' 그리고 'Standby Power'로 각각 HOT, WARM, COLD 저장 장치의 전력 소비량으로 사용하였으며 WARM 또는 COLD에 I/O가 발생하는 경우에는 HOT 영역과 동일한 전력 소모량을 가지는 것으로 가정하였다.

표 2. IBM Ultrastar 36Z15 디스크 정보
Table 2. IBM Ultrastar 36Z15 Disk Information

Standard Interface	SCSI
Individual Disk Capacity	18.4 G
Maximum Disk Rotation Speed	15000 RPM
Minimum Disk Rotation Speed	3000 RPM
RPM Step-Size	3000 RPM
Active Power(Read/Write)	13.5 W
Seek Power	13.5 W
Idle Power@15000RPM	10.2 W
Standby Power	2.5 W
Spin-up Time (Standby to Active)	10.9 secs
Spin-up Energy (Standby to Active)	135 J
Spin-down Time (Active to Standby)	1.5 secs
Spin-down Energy (Active to Standby)	13 J

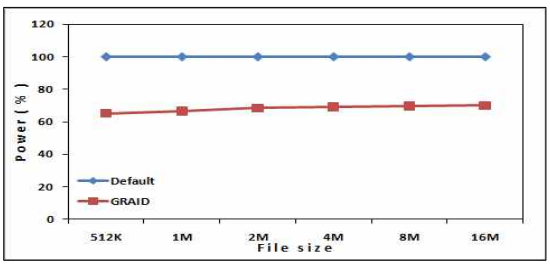


그림 10. 전력 소비량 (IOzone)
Fig. 9. Power Consumption(IOzone)

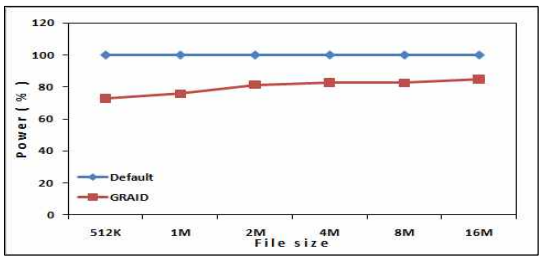


그림 11. 전력 소비량 (PostMark)
Fig. 10. Power Consumption(PostMark)

그림 9과 그림 10은 전력 예측 소비량을 표현한 것으로 앞에서 언급한 세 단계 전력량을 사용하여 HDD (Default)의 전력 사용량을 100%로 두고 G-RAID의 전력 값을 비율로 표현하였다. 그림 10의 PostMark 워크로드 전력 소비 감소율은 IOzone 워크로드보다 낮은 것으로 나타났다. 이는 PostMark 워크로드의 COLD 영역에 대한 높은 접근 빈도수가 디스크 상태를 활성화시킴으로써 발생하는 추가적인 전력 소비 때문인 것으로 분석된다.

3. 응답시간

그림 11은 disksim을 이용하여 측정한 평균 응답 시간으로 IOzone 워크로드를 사용하였다. 파일 사이즈가 4Mbytes 이상인 구간에서 일반 하드 디스크와는 다르게 상당히 높은 응답시간을 보이는 것으로 나타났으며 이는 '블록 계층화 작업'으로 인해 블록 연속 배치가 효율적으로 이루어지지 못한 결과로 분석된다.

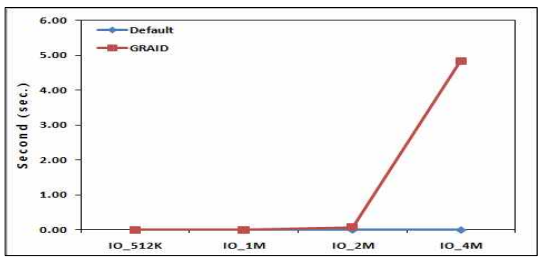


그림 12. 평균 응답 시간
Fig. 11. Average Response Time

그림 12는 그림 11에서 나타난 현상을 분석하기 위한 응답 시간 분포 도로 X축에 표현된 시간 이하의 응답 시간을 가지는 I/O 요청 수를 Y축에 나타낸다. 상위에 위치한 일반 하드 디스크의 그래프는 대부분의 I/O 요청이 5ms 이하의 응답 시간에 분포되어 있는 것에 반해서 하위에 위치한 G-RAID의 그래프는 다양한 각 I/O 요청에 대한 응답 시간이 다양하게 분포되어 있는 것을 볼 수 있다. 이는 파일 시스템에 의해 연속적으로 배치 된 블록 위치를 전력 소모량 감소를 위해 재 할당 하여 발생하는 문제점으로 분석된다. 앞으로 이 부분에 대한 개선 작업이 필요할 것으로 판단된다.

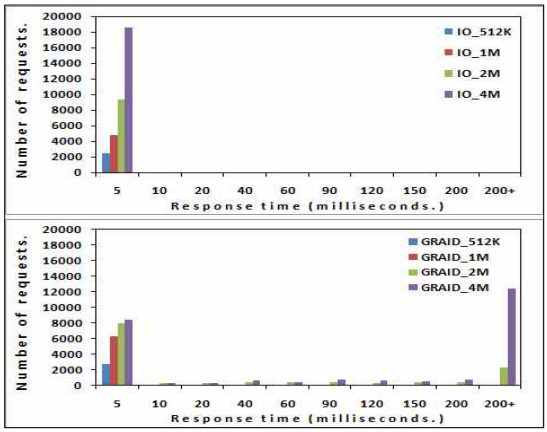


그림 13. 응답 시간 분포도
Fig. 12. Response Time Distribution

V. 결론

최근 폭발적으로 증가하고 있는 데이터센터 내의 저장장치와 서버의 전력 사용량을 절감하기 위한 다양한 연구가 진행되고 있다. 본 논문 또한 이와 같은 문제를 해결해 보고자 대용량 저장장치 시스템에서 다수의 볼륨을 계층화하는 G-RAID 기법을 제안하였다.

G-RAID는 데이터 I/O 성능 저하의 발생을 최소화하면서 전력 소비를 최소화하기 위해 저장 장치의 전력 소비량을 계층화하고 데이터 접근 빈도수에 따라 저장 장치를 선택하는 방식을 사용했다. 또한 각 계층 별 데이터 이중화 강도를 달리함으로써 저장 장치 공간을 효율적으로 사용하면서 데이터 안정성을 확보했다. 그리고 G-RAID 구현을 통한 성능 평가 결과로는 블록 재배치로 인한 일부 성능에 문제가 있지만 저장장치의 소비전력을 최대 38%까지 줄임으로서 상쇄될 수 있을 것으로 생각된다.

향후 본 논문과 관련하여 이전 성능평가에서 문제가 되었던 '블록 재배치 작업'으로 인한 응답 속도의 저하 현상을 개선하기 위해 효율적인 '재배치 작업' 및 '접근 빈도수 연산'에 관한 연구와 HDD의 대기 시간을 최소화하기 위한 '데이터 연속성 최대화'에 대한 연구를 진행할 계획이다.

참고문헌

[1] U.S. Environmental Protection Agency, "Report to Congress on Server and Data Center Energy Efficiency", Public Law 109-431, 2007

[2] C. Weddle, M. Oldham, J. Qian, A.-I. A. Wang, L. Reiher, and G. H. Kuenning. "PARAID: A Gear-Shifting Power-Aware RAID.", Proceedings of USENIX Conf. on File and Storage Tech, pp. 245-260, 2007.

[3] Dennis Colarelli, Dirk Grunwald, "Massive Arrays of Idle Disks For Storage Archieves.", Proceedings of ACM/IEEE conference on Supercomputing02, pp.1-11., 2002.

[4] Ziliang Zong, Matt Briggs, Nick O'Connor, Xiao Qin, "An Energy-Efficient Framework for Large-Scale Parallel Storage Systems", Proceedings of IPDPS'07, 2007.

[5] Eduardo Pinheiro, Ricardo Bianchini, "Energy Conservation Techniques for Disk Array-Based Servers', Proceedings of ICS"04, 2004.

[6] J.W. Um, "Green IT realized with Virtualization", Oracle Korea Magazine, pp.6-16, 2009.

[7] Manish Malhotra, "Reliability Analysis of Redundant Arrays of Inexpensive Disks", Proceedings of Journal of parallel and distributed computing, pp.146-151., 1993.

[8] J. S. Bucy, G. R. Ganger. and et al. "The DiskSim Simulation Environment Version 4.0 Reference Manual."http://www.pdl.cmu.ecu/Disksim.

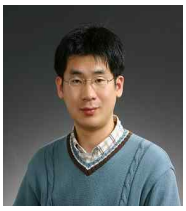
[9] blktrace(8) linux man page. http://linux.die.net/man/8/blktrace.

[10] IOzone 3.257, http://www.iozone.org, Accessed January 2006.

[11] Jeffrey Katcher, 'PostMark: A New File System Benchmark.' http://www.netapp.com/technology/level3/3022.html

[12] Qingbo Zhu, Asim Shankar and Yuanyuan Zhou, "'PB-LRU: A Self-Tuning Power Aware Storage Cache Replacement Algorithm for Conserving Disk Energy", Proceedings of ICS'04, 2004.

저 자 소 개



김 영 환

2001 : 부경대학교 컴퓨터공학과 공학사.
2003 : 성균관대학교 컴퓨터공학과 공학석사.
2009 : 숭실대학교 컴퓨터공학과 공학박사 수료
현 재 : 전자부품연구원 융합신산업연구본부 선임연구원
관심분야 : 저장장치, 임베디드 시스템
Email : yhkim93@keti.re.kr

석 진 선

2005 : 세종대학교 소프트웨어공학과 공학사.
2007 : 세종대학교 컴퓨터공학과 공학석사.
2010 : 세종대학교 컴퓨터공학과 공학박사 수료
현 재 : 전자부품연구원 융합신산업연구본부 전임연구원
관심분야 : 파일시스템
Email : durgatm@gmail.com

박 창 원

1986 : 중앙대학교 전자공학과 공학사.
2002 : 광운대학교 전자통신공학과 공학석사.
2007 : 성균관대학교 컴퓨터공학과 공학박사 수료
현 재 : 전자부품연구원 융합신산업연구본부 수석연구원
관심분야 : 저장장치, 헬스케어
Email : parkcw@keti.re.kr

홍 지 만

1994 : 고려대학교 전산학과 공학사.
1997 : 서울대학교 컴퓨터공학과 공학석사.
2003 : 서울대학교 컴퓨터공학과 공학박사
현 재 : 숭실대학교 컴퓨터학부 부교수
관심분야 : 운영체제, 임베디드 시스템
Email : jiman@ssu.ac.kr