

# 하이브리드 하드디스크에서 AHP를 적용한 블록 교체 기법

## (A Block Replacement Scheme using Analytic Hierarchy Process in Hybrid HDD)

김정원<sup>1)\*</sup>  
(Jeong-Won Kim)

**요약** 하이브리드 하드디스크는 저전력이면서 마그네틱 하드디스크에 비해 읽기 성능이 우수하다. 이 디스크의 플래시메모리에는 지역성이 높은 블록이 저장되므로 효율적인 블록 교체 기법이 필요하다. 블록 교체에는 크기, 지역성, 빈도 등 다양한 요인에 의해 결정되므로 일종의 다중 기준 결정 모델로 정의될 수 있다. 이 문제를 해결하기 위해 본 연구는 AHP (Analytic Hierarchy Process) 모델을 적용하여 효율적인 블록 교체 기법을 제시한다. 실험의 효율성을 검증하기 위하여 철저한 시뮬레이션을 수행한 결과 일반 응용 프로그램의 응답성뿐만 아니라 부트 시간이 단축됨을 확인하였다.

**핵심주제어** : 블록 교체, 분석적 계층 과정, 하이브리드 하드 디스크

**Abstract** The read performance of hybrid hard disk is better than the legacy hard disk and power consumption is also considerably low. As blocks with enough localities may be located in the non-volatile cache whose size is generally limited, an effective block replacement scheme is required. As this replacement is inevitably affected by various parameters, we define this issue as a kind of multiple criteria decision model. To solve this problem, this paper suggests a new block replacement algorithm based on the analytic hierarchy process. Through simulation for our model, we confirmed that the proposed model could be used as a replacement algorithm of the hybrid hard disk as it may improve boot time as well as response time of general applications.

**Key Words** : Block Replacement, AHP, Hybrid hard disk

### 1. 서론

SSD(Solid-state drive)는 강력한 입출력 기능뿐만 아니라 저전력 기능을 제공하여 저장장치로 각광을 받고 있으나 전통적인 마그네틱 디스

크는 아날로그적 기술과 기계적 성능에 기반하므로 사용자의 만족도가 낮다[1]. 그러나 이 마그네틱 디스크는 비용과 운영체제 친화성으로 인해 대용량 저장 장치에는 현재까지도 이용되고 있고 순차적 접근을 요구하는 응용에는 좋은 성능을 제공한다.

한편, SSD의 장점에도 불구하고 용량의 제한과 고가의 특성으로 인해 HDD의 수요에 미치지 못하고 있다. 이러한 점을 개선하기 위하여

\* Corresponding Author : jwkim@silla.ac.kr

Manuscript received July 10, 2015 / revised October 14, 2015  
/ accepted October 28, 2015

1) 신라대학교 컴퓨터정보공학부

몇몇 회사들은 마그네틱 디스크와 플래시 메모리의 장점을 결합한 하이브리드 디스크(HHD : Hybrid hard disk) 를 출시하였다[2]. 이 디스크의 목표는 저장장치의 입출력 처리율을 향상시키는 것인데 디스크 내에 낸드 플래시 메모리를 장착하여 지역성 높은 블록을 고정시키고 읽기 속도를 개선시키는 것이다. 즉 HDD 의 대용량 공간과 플래시 메모리의 읽기 속도의 장점을 결합한 것이다[3,4,5]. 이 HHD 에는 애플리케이션의 입출력에 영향을 미치지 않으면서 플래시 메모리와 마그네틱 디스크 사이의 블록 전송 문제, 플래시 메모리에 캐싱 방법, 전력 실패에도 불구하고 데이터 전송의 신뢰성 해결 등의 문제가 존재한다[6].

HHD 의 낸드 플래시 메모리에 지역성 높은 부트 프로그램, 애플리케이션, 그리고 데이터를 고정한다면 뛰어난 접근 속도를 제공할 수 있다. 그러나 운영체제는 이들 데이터를 인식하지만 하드디스크는 독립성으로 인해 인식하지 못한다. 따라서 하드디스크는 이 블록들에 대한 접근 빈도, 이력, 그리고 범위 등의 패턴을 유지해야 하는데 전통적인 LRU, MRU, Aging 기법 등이 적용될 수 있으나 이상적 해결책은 존재하지 않는다. 일반적으로 NVC(Non-volatile cache) 와 플래터 사이의 블록 교체 기법에 통계적 기법들이 이용되고 있는데 사용자는 응답시간을 중시하고 서비스 제공자는 부하 균등, 처리율을 중시하므로 본 논문에서는 이 문제를 일종의 다중 기준 결정문제(MCDM : multi-criteria decision making) 로 정의한다[7].

이 MCDM 을 해결하기 위해 본 연구에서는 AHP (Analytic Hierarchy Process) 기법을 도입하는데 다양한 대안 중에서 최선의 해결책을 선택하는 기법이다. 각 대안들은 선택 기준 또는 특성을 가지며 응용의 목적에 따라서 다양한 가중치를 부여한다. 각 대안에서 응용이 요구하는 특성을 만족시키기 위한 가중치를 적용하면 가장 높은 점수를 받은 대안이 선택되는 기법이다. 따라서 AHP 를 HHD 에 적용하면 접근빈도, 에이징, 임의성 등은 특성이 되며 디스크 블록 그룹은 대안이 되며 목적은 응용이 요구하는 블록을 NVC 에 최대한 오래 유지하는 것이다. 따라서

본 논문은 NVC 와 디스크 사이의 블록 교체에 AHP 모델을 적용한 새로운 기법을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구와 관련된 기존 연구들을 분석하고 3장에서는 AHP 모델에 기반한 블록 교체 기법을 제시한다. 4장에서는 본 연구의 효율성을 검증하기 위해 실시한 실험 결과를 설명하며 5장에서 본 연구의 기여와 결론을 제시한다.

## 2. 관련 연구

지역성 높은 블록을 NVC 에 복사하는 것은 일종의 선반입 기법으로 유닉스 계열의 운영체제에서는 메인 메모리로 복사하는데 리눅스에서는 Read-ahead 로 불린다. 이 NVC 를 플래시 메모리로 활용하는 기법에는 가까운 미래에 사용될 가능성이 높은 블록들을 미리 복사하여 플래시 메모리를 일종의 캐싱 영역으로 간주하고 HDD 의 스핀들 모터를 가급적 오랜 기간 동안 정지시키는 기법들이 제안되었다[8,9].

애플리케이션이 병렬로 블록에 접근하는 경우 블록 요구 순서가 반복되는 패턴을 이용하여 블록에 순위와 거리의 조합을 이용하여 우선순위가 높은 순서로 NVC 에 캐싱하기도 한다[10]. 그리고 메인 보드에 플래시 메모리를 두고 블록을 선반입하여 마그네틱 플래터의 접근 빈도를 낮추고 플래시 메모리의 히트율을 높이는 EXCES (External caching in energy saving storage system) 기법이 제안되었으나 비용이 높은 측면이 있다[11].

이 외에도 운영체제가 블록 접근 패턴을 미리 HHD 에 알려주는 기법이 제안되었으나 저장장치의 독립성 문제로 실효성이 낮은 것으로 보이며 플래시 메모리를 읽기/쓰기 영역으로 구분하고 모터가 정지되면 플래시메모리로 입출력을 리다이렉트하는 기법들이 제안되었으나 주로 전통적인 블록 교체 알고리즘이 사용되었다[12]. 본 연구는 기존의 기법이 아니라 다중 기준을 적용하고 애플리케이션의 요구에 부합하는 대안을 선택하는 기법을 제시한다.

### 3. AHP 에 기반한 블록 교체 기법

#### 3.1 시스템 구성

본 연구의 HDD 는 DRAM cache, NVC, 그리고 플래터 세 가지 구성된다고 가정한다. 먼저 읽기의 경우 DRAM cache 에 히트되지 않으면 NVC 에서 검색한다. NVC 에도 히트되지 않으면 최종 플래터에서 블록을 전송한다. 제안하는 알고리즘 적용을 위해서 DRAM cache 에는 캐싱 정책을 적용하지 않으며 NVC 적중의 경우 LBA(Logical Block Address), 빈도, 에이징, 임의도 등의 정보가 저장된다. 쓰기 연산의 경우 NVC 적중 여부를 검색하며 NVC 에 존재하지 않고 여유 공간이 있으면 NVC 에 바로 저장한다. 이것은 스핀 모터의 동작을 요구하지 않으므로 전력을 아낄 수 있으며 스핀 모터가 동작하고 있어도 쓰기 연산의 속도가 향상될 수 있다. 만약 공간이 부족할 경우 논문의 알고리즘에 의해 교체 대상 블록을 선정하여 플래터로 보내고 빈 자리에 블록을 복사한다.

#### 3.2 블록 교체 기법

일반적으로 AHP 는 오브젝트 레벨, 속성 레벨, 그리고 대안 레벨로 구성된다. 오브젝트 레벨에서는 다양한 파라미터에 가장 잘 부합하는 LBA 그룹을 선택하는 것이다. 속성 레벨은 접근 빈도, 에이징, 그리고 임의성 등의 결정 기준에 따라 대상 블록에 대한 교체의 순위를 정한다. 마지막으로 대안 레벨은 캐싱 또는 교체 대상 LBA 그룹을 나타낸다. 이것을 HDD 에 적용하면 속성 레벨의 결정 기준에 가중치를 부여하여 특정 블록 그룹을 교체대상으로 선택할 수 있다.

AHP 모델에서 결정기준은 일반적으로 우선순위를 가지는데 본 연구에서는 1에서 9까지의 값으로 정하고 9가 가장 높은 우선순위임을 의미한다. 이 값은 LBA 그룹의 인기도를 표현하고 계산하는데 사용되며 빈도, 에이징, 임의성과 같은 결정기준의 파라미터를 표현한다. NVC 에 캐싱된 LBA 그룹의 집합을  $\{G_1, G_2, \dots, G_m\}$  로 결정

기준의 집합을  $\{C_1, C_2, \dots, C_n\}$  으로 표현한다. 교체 대상의 선택 과정은 두 단계로 진행되는데 첫 번째는 쌍비교에 의해 가중치 벡터를 계산하고 속성, 대안 레벨에서 일관성 검사를 한다. 두 번째 단계에서는 각 레벨에서의 가중치 벡터를 곱하여 최선의 그룹을 선택하는데 자세한 과정은 다음과 같다.

단계 (1) : 오브젝트 레벨과 속성 레벨사이의 가중치 행렬, 일관성 지수 계산

오브젝트 레벨과 속성 레벨사이의 쌍비교 행렬을 PWCM(Pair Wise Comparison Matrix) 라고 하고 이는 모든 결정 기준에 대한 LBA 그룹의 선호도를 의미한다. PWCM 이 식 (1)처럼  $n \times n$  행렬이고 PWCM( $i, j$ ) 의 요소가 5이면 PWCM( $j, i$ ) 의 값은 5의 역수인 1/5이 된다.

$$PWCM^{i,j} = \begin{cases} 1 & \text{if } i = j \\ \frac{1}{PWCM^{j,i}} & \text{else} \end{cases} \quad \text{식 (1)}$$

$$\begin{bmatrix} \frac{w_1}{w_1} & \dots & \frac{w_1}{w_n} \\ \frac{w_1}{w_1} & \dots & \frac{w_1}{w_n} \\ \vdots & \ddots & \vdots \\ \frac{w_n}{w_1} & \dots & \frac{w_n}{w_n} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = \lambda_{\max} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \quad \text{식 (2)}$$

만약  $n$  개의 결정기준의 의해 생성된 쌍비교 행렬에 대해 우선순위 벡터를 계산하는데 식 (2)를 계산하면 된다. 여기서  $\lambda_{\max}$  는 PWCM 의 고유값이며  $w^{criteria}$  로 표현된다. PWCM 의 가능한 조합의 계산으로 가중치 벡터를 얻을 수 있는데 본 연구에서는  $w^{criteria} = \{w_1, w_2, \dots, w_n\}$  으로 정의한다. 이 모델에서 핵심 과정은 식 (3)으로 가중치 벡터를 계산하는 것이다. 이 양의  $n \times n$  행렬은  $PWCM = (w_i/w_j)$  의 비로 표현되며  $n$  이 고유값이고 PWCM 과  $w^{criteria}$  의 곱이  $\lambda_{\max} \cdot w^{criteria}$  과 같으면 일관성이 있으며 만약  $w^{criteria}$  이 0보다 크면 유일하다. 이 일관성 비율(CR : Consistency Ratio) 은 식 (4)로 표현된다. 식 (4)에서 RI 는 랜덤 지수로 비교 행렬의

우선 순위에 의해 임의적으로 정하는데 CR 값이 0.1 보다 작으면 PWCM 은 일관성이 있음을 의미한다.

$$PWCM \times w^{criteria} = \lambda_{max} \times w^{criteria} \quad \text{식 (3)}$$

$$CR = \frac{CI}{RI}, \text{ where } CI = \frac{\lambda_{max} - n}{n} \quad \text{식 (4)}$$

단계 (2) : 속성 레벨과 대안 레벨사이의 가중치 벡터와 일관성 지수의 계산

다음 단계는 모든 대안에 대해서 각 결정 기준의 가중치 벡터와 일관성 지수를 계산하는 것이다. 식 (1), (2), (3), 그리고 (4)를 이용하여 제안하는 알고리즘은 결정기준에 대하여 LBA 그룹 각각에 대한 가중치 행렬과 일관성 비율을 계산한다. 각 그룹의 가중치 벡터가  $w^{group} = \{w_1, w_2, \dots, w_n\}$  이라 가정하고 이는  $m \times n$  행을 구성한다. 이때  $m$  은 그룹의 수이고  $n$  은 결정 기준의 수이다. 알고리즘은 단계 (1), 단계 (2)에서 계산한 가중치 벡터를 곱하여 각 블록 그룹에 대한 득점을 식 (5)에서처럼 계산한다. 이 벡터에서 최소값을 가진 LBA 그룹이 교체 대상으로 선정된다. 최대값은 목적에 가장 부합하는 블록이므로 NVC 에 오래동안 유지시켜 캐쉬 적중율을 높여야 한다.

$$\begin{aligned} score &= w^{group} \times w^{criteria}, \\ \text{where } w^{group} &\text{ is } m \text{ by } n \\ \text{and } w^{criteria} &\text{ is } n \text{ by } 1 \end{aligned} \quad \text{식 (5)}$$

### 3.3 블록 교체 기법의 예

AHP 모델을 사용하여 교체 대상 블록 그룹을 선택하는 예를 표로서 보여준다. Table 1 은 결정

Table 1 Example of preference matrix

	frequency	aging	randomness
frequency	1	1/3	5
aging	3	1	7
randomness	1/5	1/7	1
sum	21/5	31/21	13

기준에 대한 선호도를 표현한 것이다.

아래는 쌍비교 행렬과 가중치 벡터이다. 가중치 벡터에서 에이징이 가장 높은 0.6583을 기록하고 있는데 이것은 태스크가 NVC에 오래 머문 블록을 더 자주 이용한다는 것을 의미한다.

$$PWCM = \begin{bmatrix} 5 & 7 & 5 \\ 21 & 31 & 13 \\ 15 & 21 & 7 \\ 21 & 31 & 13 \\ 1 & 3 & 1 \\ 21 & 31 & 13 \end{bmatrix}, \text{ weight Vector} = \begin{bmatrix} 0.2657 \\ 0.6583 \\ 0.0759 \end{bmatrix}$$

CR을 계산하기 위해서 아래에서 보듯이 가중치 벡터의 고유값에 우선순위 행렬을 곱한다. 식 (3), (4)를 사용하면  $\lambda_{max}$  가 3.0967, CR 이 0.047 이다. CR 값이 0.1 보다 작으므로 우선순위 행렬은 일관성이 있다고 판단할 수 있다.

$$\begin{aligned} \lambda_{max} &= \frac{5}{21}(0.2828) + \frac{31}{21}(0.6434) + 13(0.0738) \\ &= 3.0967 \end{aligned}$$

Table 2  $\lambda_{max}$  and CR for Frequency

LBA	$G_1$	$G_2$	$G_3$	$G_4$	$G_5$	weight
$G_1$	1	4	1/6	1/5	1/7	0.0785
$G_2$	1/4	1	1/5	3	2	0.1799
$G_3$	6	5	1	6	3	0.6462
$G_4$	5	1/3	1/6	1	2	0.1532
$G_5$	7	1/2	1/3	1/2	1	0.0827
$\lambda_{max} = 7.3997, CR = 0.0480 (< 0.1)$						

Table 3  $\lambda_{max}$  and CR for Aging

LBA	$G_1$	$G_2$	$G_3$	$G_4$	$G_5$	weight
$G_1$	1	2	1/3	3	1/5	0.1303
$G_2$	1/2	1	1/2	1/2	1/2	0.0475
$G_3$	3	2	1	4	1/4	0.2348
$G_4$	1/3	2	1/4	1	1/3	0.1168
$G_5$	5	2	1	3	1	0.2166
$\lambda_{max} = 6.1495, CR = 0.0230 (< 0.1)$						

Table 4  $\lambda_{max}$  and CR for Randomness

LBA	$G_1$	$G_2$	$G_3$	$G_4$	$G_5$	weight
$G_1$	1	6	1/3	2	1/4	0.0159
$G_2$	1/6	1	1/3	2	1/3	0.0248
$G_3$	3	3	1	5	2	0.1034
$G_4$	1/2	1/5	1/5	1	1/5	0.0658
$G_5$	4	1/2	1/2	5	1	0.5263
$\lambda_{max}=55698, CR= 0.0114(<0.1)$						

Table 2, 3, 4에서는 5개의 LBA 그룹에 대하여 결정기준에 대한 고유값과 일관성, 그리고 가중치 벡터를 계산한 것이다. 각 결정기준에 대하여 모든 CR 이 0.1 보다 작으므로 선호도는 일관성이 있다고 가정할 수 있다. 마지막으로 각 LBA 그룹의 점수는 식 (5)에 의하여 대안 레벨에서의 가중치 행렬에 속성 레벨의 가중치 벡터를 곱하여 계산한다. 아래의 예는 각 그룹의 점수 벡터이다. 아래의 예에서는 그룹 2가 가장 낮은 점수인 0.0809를 획득하였으므로 교체대상으로 선정됨을 알 수 있다.

$$score = w^{group} \times w^{criteria} =$$

$$\begin{bmatrix} 0.07850 & 0.13030 & 0.0159 \\ 0.17990 & 0.04750 & 0.0248 \\ 0.64620 & 0.23480 & 0.1034 \\ 0.15320 & 0.11680 & 0.0658 \\ 0.08270 & 0.21660 & 0.5263 \end{bmatrix} \times \begin{bmatrix} 0.2657 \\ 0.6583 \\ 0.0759 \end{bmatrix} = \begin{bmatrix} 0.1078 \\ 0.0809 \\ 0.3341 \\ 0.1226 \\ 0.2045 \end{bmatrix}$$

#### 4. 실험결과

제안하는 기법의 성능 평가를 위해 호스트와 HDD 모델을 위한 실험 환경을 구축하였다. 호스트 모듈은 모든 입출력 명령을 생산하고 NVC에 저장된 블록 그룹에 대한 기록을 유지한다. HDD 모듈은 실제 하드디스크를 시뮬레이션하며 NVC, 마그네틱 디스크, 그리고 DRAM 으로 구성된다. 모든 시뮬레이션은 매트랩과 시뮬링크상에서 구현되었으며 디스크는 Segate 8GB MLC, 5,200RPM, 2TB, SATA 인터페이스를 모델링하였다.

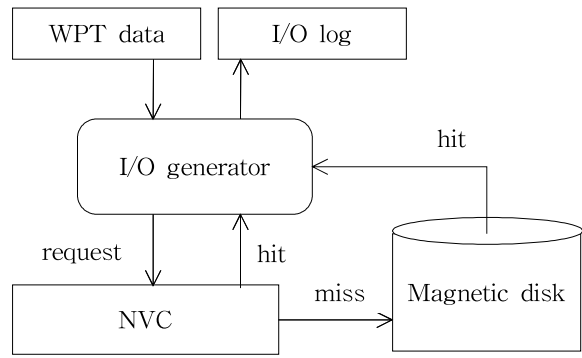


Fig. 1 Simulation model

실제와 유사한 환경을 구축하기 위해서 Winodws Performance Toolkit 을 통해 생산되는 부트 과정 및 애플리케이션들의 입출력 패턴을 데이터로 사용하였다. 두 가지 종류의 패턴이 사용되었는데 부트 과정에서의 패턴과 웹 브라우징, 문서 작업, 멀티미디어 편집 등의 일반 애플리케이션들이다. 이들 패턴들은 4주간의 평균값을 사용하였다. Fig. 1은 시뮬레이션 모델이고 Table 5는 실험에 사용된 입출력 패턴을 정리한 것이다.

Table 5 Details of Boot and General Application traces

	boot trace	general application
read count	2,856	45,673
write count	870	27,539
data size	1,231,889 secotors	read : 10,557,980 write : 4,346,745

실험에서 사용된 HDD 의 디스크 접근 시간은 NVC 와 플래터 접근 시간으로 구성된다. NVC 의 접근 시간은 그룹의 크기에 비례하므로 쉽게 계산되나 플래터의 접근 시간은 기계적 성질에 의해 좌우되므로 탐색시간, 회전지연시간, 그리고 전송시간으로 구성된다. 따라서 LBA 그룹에 속한 모든 섹터를 전송하기 위해서는 다음과 같은 식 (6)으로 결정된다.

$$Access\ Time = T_{NVC} + T_{seek} + T_{rotationaldelay} + T_{LBA} \quad \text{식 (6)}$$

본 실험에서는 호스트 모듈이 셧다운되기 전에 모든 부트 데이터가 NVC 에 고정되며 호스트가 부트될 때 입출력은 연속성에 기반하여 NVC 또는 플래터에서 수행된다. LBA 그룹 크기가 0인 것은 NVC 에 데이터가 고정되지 않는 상황 즉, 플래터에서 모든 입출력이 이루어짐을 의미한다. HHD 의 부트 시간은 전통적 디스크보다 낮는데 전통적 디스크는 LBA 그룹 크기가 다양하더라도 캐시에 고정된 데이터가 없기 때문이다. 또한 모든 디스크가 DRAM 으로 블록을 캐싱하지만 크기가 작아 전체적인 성능에는 영향을 거의 미치지 못한다.

Fig. 2에서 보듯이 LBA 그룹 크기가 증가할수록 부트시간도 비례하여 향상되는데 모든 부트 블록을 NVC 에 고정시키면 최대의 성능을 발휘할 수 있을 것이다. 하지만 NVC 의 용량이 커지면 비용이 높아지고 일반 응용을 위한 공간이 부족하게 되어 적절한 수준의 데이터 고정이 필요하다.

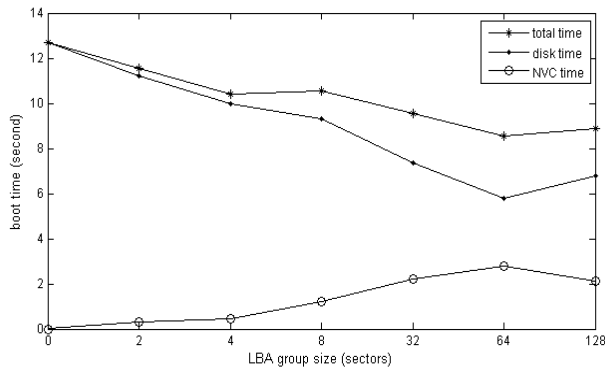


Fig. 2 Average boot time vs. LBA Group size

일반 응용 프로그램은 다양한 실행 모듈 또는 라이브러리로 구성된다. 이들이 실행되기까지의 총 시간은 정적, 동적 라이브러리, 실행 코드 등이 적재되는데 필요한 시간들로 구성된다. 이들 모듈의 크기를 미리 파악하기는 어려우므로 본 실험에서는 실행 코드는 전체 파일의 50%에서 90%로 간주하고 정적 또는 동적 라이브러리는

10%에서 50%로 계산하였다. HHD 제조사들은 NVC 의 블록 교체 기법은 통계적 방법으로 처리한다고 밝히고 있지만 정확한 기법을 알기 어려워 제안하는 기법과 직접적인 비교는 어렵다.

따라서 본 논문에서는 전통적인 기법인 MRU (Most Recently Used) 와 MFU(Most Frequently Used) 기법과 제안하는 기법을 비교한다. 이 기법을 선택한 이유는 HHD 에서 블록 교체에 가장 큰 영향을 미치는 것이 접근 빈도와 에이징 속성이기 때문이다. 알려진 것처럼 MRU 는 에이징에 기반한 방법이며 MFU 는 접근 빈도에 기반하므로 본 연구의 기법과 좋은 비교 대상이다.

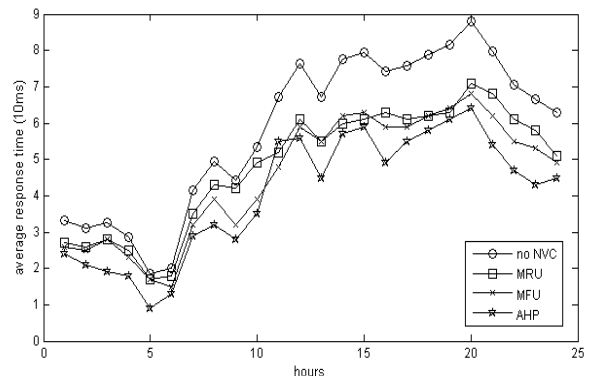


Fig. 3 Average response time vs. Time

Fig. 3은 시간에 따른 응용프로그램의 평균 응답시간을 나타낸다. 실험의 초기에는 모든 기법들이 실험의 중반에 비해 낮은 응답 시간을 보인다. 이것은 아직 충분한 블록들이 NVC 에 고정되지 않았기 때문이다. 당연히 NVC를 사용하지 않는 기법보다는 높은 성능을 보이고 있다. 실험의 중반에 도달하면 응답 시간이 빨라지는데 이것은 많은 블록들이 NVC에 복사되고 일부는 고정되기 때문이다. 그러나 실험의 후반으로 갈수록 모든 기법들의 응답시간이 증가하는데 이것은 다양한 응용들이 다양한 데이터를 요구하므로 요구 블록들이 랜덤하게 흩어져 있고 이들 블록들이 NVC 에 기존 블록들과 교체 경쟁을 하기 때문으로 분석된다.

그리고 본 논문에서 제안하는 기법이 MRU, MFU 에 비해 나은 성능을 보이는데 MFU 는 접근 빈도만 고려하기 때문에 고정된 시간이 오

래되더라도 사용될 가능성이 높은 블록들로 존재하며, MRU 의 경우 에이징만 고려하여 부트 블록과 같이 자주 사용되는 블록들이 교체되어 NVC miss 발생 확률이 높기 때문인 것으로 분석된다. 반면 AHP 의 경우 응용의 특성에 따라 접근 빈도, 에이징, 그리고 임의성까지 모두 고려하므로 두 기법에 비해 나은 성능을 보인다.

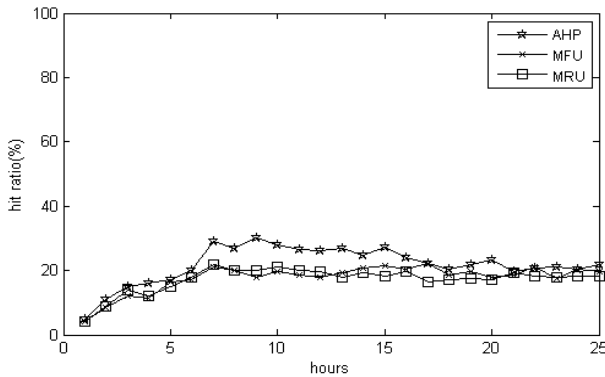


Fig. 4 Hit ratio vs. time

Fig. 4는 시간에 따른 NVC 히트율을 나타내고 있다. 실험 초기에는 세 기법 모두 낮은 히트율을 보이고 있는데 이것은 아직 NVC에 캐시가 올라오지 않았기 때문이다. 그러나 시간에 지날수록 세 기법 모두 20% 이상의 캐시 히트율을 보이고 있다. 실험 초기에 히트율은 전반적으로 상승하지만 10시간 정도 지나면 모든 기법의 히트율이 증가하지 않는데 이것은 NVC 에 적재되는 블록의 요구 패턴이 안정화되기 때문이다. Fig. 3의 실험에서와 마찬가지로 논문에서 제안하는 기법이 MRU, MFU 에 비해 높은 히트율을 나타내고 있다. 이것은 Fig. 3의 평균 응답시간 분석에서와 같은 이유로 판단된다.

## 5. 결론

본 논문에서는 하이브리드 하드디스크를 위한 새로운 블록 교체 기법을 제안하였다. 이 기법은 블록이 NVC 에 고정되고 플래터로 교체될 때 적용되는 알고리즘이다. 데이터를 고정하고 교체할 때 다양한 요인을 고려해야하는데 이 문제를

본 논문에서는 다중 결정 기준 문제로 정의하며 AHP 를 사용한 모델을 제안하였다. 전통적 블록 교체 기법은 접근 빈도, 에이징 등 일부 기준만 사용하는데 비해 제안 모델은 애플리케이션의 특성에 따라 접근 빈도, 나이, 그리고 임의성까지 동시에 고려하여 최적의 블록을 교체하도록 하였다. 제안 기법의 성능 평가를 위해 다양한 실험을 진행하였으며 전통적인 기법에 필적하는 성능을 보이고 있어 HDD 를 위한 새로운 블록 교체 기법으로 적용될 수 있음을 확인하였다.

## References

- [1] R.Panabaker, "Hybrid hard disk & ReadyDrive technology: improving performance and power for Windows Vista mobile PCs", In Proceeding of Microsoft WinHEC, 2006. (conference)
- [2] D. Reinsel and J. Rydning, "HohnBreaking the 15K-rpm HDD Performance Barrier with Solid State Hybrid Drives," IDC, No. 244250, 2013. (white paper)
- [3] Y. J. Song and J. Y. Lee, "A study of external device file outflow", Journal of the Korea Industrial Information System Society, Vol. 16, No. 2, pp. 59-64, 2011. (journal)
- [4] Y. J. Song and J. Y. Lee, "Forensic investigation of external USB drive", Journal of the Korea Industrial Information System Society, Vol. 15, No. 4, pp. 39-45, 2010. (journal)
- [5] Y. J. Nam, E. J. Yang, J. Y. Lee, S. R. Kim, and D. W. Seo, "Analysis of power consumption patterns for commercial portable multimedia players", Journal of the Korea Industrial Information System Society, Vol. 12, No. 3, pp. 95-103, 2007. (journal)
- [6] <http://www.datacore.com/Software/Features/List-of-Features/Auto-Tiering.aspx>
- [7] D. Benjamin and D. Bamford, "Development,

test and comparison of two Multiple Criteria Decision Analysis (MCDA) models: A case of healthcare infrastructure location”, Expert Systems with Applications, Vol. 42, No 19, pp. 6717-6727, 2015. (*journal*)

- [8] F. Chen, S. Jiang, and X. Zhang, “SmartSaver: turning flash drive into a disk energy saver for mobile computers”, In Proceeding of 11th ACM/IEEE International Symposium on Low Power Electronics and Design, pp. 412-417, 2006. (*conference*)
- [9] T. Kgil and T. Mudge, “FlashCache: A NAND flash memory file cache for low power web servers”, In Proceeding of the 2006 International Conference on Compilers, Architecture and Synthesis for Embedded Systems, pp. 103-112, 2006. (*conference*)
- [10] B. Gill and L. Bathen, “AMP: Adaptive multistream prefetching in a shared cache”, In Proceeding of the 5th USENIX Conference on File and Storage Technologies, pp. 185-198, 2007. (*conference*)
- [11] L. Useche, J. Guerra, and M. Bhadkamkar, “EXCESS: External Caching in Energy Saveing Storage Systems”, In Proceeding of the 13th International Symposium on High-Performace Computer Architecture, pp.89-100, 2008. (*conference*)
- [12] J. No, “NAND flash memory-based hybrid file system for high I/O performance”, Journal of Parallel and Distributed Computing, Vol. 72, No 12, pp.1680-1695, 2012. (*journal*)



김 정 원 (Jeong-Won Kim)

- 정회원
- 부산대학교 전자계산학과 학사
- 부산대학교 대학원 전자계산학과 석사
- 부산대학교 대학원 전자계산학과 박사
- 현재 : 신라대학교 컴퓨터정보공학부 교수
- 관심분야 : 클라우드 컴퓨팅, 내장형시스템, 멀티미디어, 운영체제, 모바일 컴퓨팅