한국산학기술학회논문지
Vol. 10, No. 8, pp. 1962-1969, 2009

NAND 플래시 메모리 기반의 대용량 저장장치 설계

류동우[1*], 김상욱[1], 맹두열[1]
[1]중앙대학교 컴퓨터공학과

Design of an Massive Storage System based on the NAND Flash Memory

Dong-Woo Ryu[1*], Sang-Wook Kim[1] and Doo-Lyel Maeng[1]
[1]Division of Computer Science & Engineering, Chung-Ang University

요 약 과거 20년 동안, 우리는 CPU, 메모리, 네트워크 장비 그리고 하드디스크를 포함한 컴퓨터의 주요 구성 요소에 대하여 눈부신 향상을 보아왔다. 용량 면에서의 굉장한 발전에도 불구하고, 컴퓨터의 구성요소들 중 하드디스크는 처리 시간이 가장 지연되는 장치이고, 가까운 미래에 이러한 문제가 해결될 것이라 예측하기 어렵다. 우리는 NAND 플래시메모리를 이용하여 이러한 문제를 해결하기 위한 새로운 접근 방법을 제시한다. 저장 수단으로서의 플래시 메모리 이용에 대한 연구는 현재 많이 이루어져왔으나, 그러한 연구의 대부분은 모바일이나 내장형 장치에 중점 되어있다. 우리의 연구는 기업 단위의 서버 시스템 까지도 아우르는 저장 시스템으로서의 NAND 플래시 메모리를 발전시키는데 목표를 두고 있다. 본 논문은 기존의 저장 시스템 기반의 NAND 플래시 메모리의 단점을 극복하기 위하여 구조적이고 운영 가능한 메커니즘을 제시하고 평가한다.

Abstract During past 20 years we have witnessed brilliant advances in major components of computer system, including CPU, memory, network device and HDD. Among these components, in spite of its tremendous advance in capacity, the HDD is the most performance dragging device until now and there is little affirmative forecasting that this problem will be resolved in the near future. We present a new approach to solve this problem using the NAND Flash memory. Researches utilizing Flash memory as storage medium are abundant these days, but almost all of them are targeted to mobile or embedded devices. Our research aims to develop the NAND Flash memory based storage system enough even for enterprise level server systems. This paper present structural and operational mechanism to overcome the weaknesses of existing NAND Flash memory based storage system, and its evaluation.

Key Words : Flash Memory, Storage System, Disk Array, Solid State Disk

1. 서론

기존의 HDD 기반 저장 시스템은 용량 및 전송속도 측면에서 급격한 발전을 이루어 왔으나 응답시간 측면에서는 거의 발전이 없다시피 하고 있다(표 1 참조). 지난 25년간의 컴퓨터 시스템 구성 요소들의 성능향상을 고려할 때, HDD의 주요 성능들은 상대적으로 수백배에서 수천배의 퇴보를 이루었다고 보아도 과언이 아니다.

[표 1] HDD 성능 향상 추세

항목	1981	1989	2005
avg. seek time	30㎳	12.5㎳	3.6㎳
avg. latency	8.3㎳	7.0㎳	2.0㎳
transfer rate	806KB/s	1.7MB/s	~96MB/s

이러한 문제를 해결하기 위해서 S/W및 H/W 양 측면에서 무수한 연구가 이루어져 다소의 성과(RAID, JFS,

*교신저자 : 류동우(rdw2486@naver.com)
접수일 09년 07월 14일 수정일 (1차 09년 08월 07일, 2차 09년 08월 13일) 게재확정일 09년 08월 19일

Cache, Prefetch) 를 거두기는 하였으나 근본적인 원인을 해결하지 못하고 있다. 그 근본적인 원인이란 바로 HDD의 본질적 특성인 기계적 운동부의 구동속도 한계이다.

이를 해결하기 위해 다양한 연구가 이루어지고 있으나 현재 가장 두각을 나타내고 있는 HDD 대체물은 바로 플래시 메모리 기반의 저장장치이다.

플래시 메모리는 가전, 통신기기, 휴대기기 등 다양한 기기의 저장매체로서 사용되고 있으며 최근, 내장형 기기들이 보다 다양한 기능을 제공하고, 대용량 멀티미디어 데이터에 대한 수요가 증가하면서 플래시 메모리가 대용량 저장매체로서 각광받고 있다. 비휘발성의 특성을 가진 반도체 저장매체인 플래시 메모리는 집적도가 높고 낮은 소비 전력으로 구동이 가능하다. 또한, 빠른 읽기/쓰기 성능을 보이며 온도와 충격에 대한 강한 내구성을 가지고 있기 때문에, 불안정한 환경에서 동작해야 하고, 그 크기와 전원공급이 제한된 내장형 기기의 저장매체로도 매우 적합하다. 이러한 플래시 메모리의 특성 중, 서버 시스템의 저장장치로서 주목할 만한 특성을 HDD와 비교하면 다음과 같다.

HDD에 비교적 근접한 용량/공간을 제공한다.
기계적 운동부가 없다.
응답속도가 빠르다. (HDD 대비 50～100배 빠름)
단일칩만 사용할 경우에는 전송속도가 느린편이다. (HDD 대비 3~10배 느림)
용량대비 가격이 높은 편이다. (HDD 대비 50배 높음)

위와 같은 특성을 가진 플래시 메모리에 기반한 저장장치를 제작할 경우 비용 측면에서는 기존 HDD에 비할 수 없지만 빠른 응답속도를 얻을 수 있다. 그러므로 플래시 메모리 기반 저장장치의 가치는 용량대비/비용 측면보다 응답시간/비용 측면이 더욱 중요한 분야의 경우에 매우 높아진다고 볼 수 있다. 최근 국내 정보 산업계에서 응답시간/비용 측면이 부각되고 있으며 다음과 같이 예를 들 수 있다.

휴대인터넷 : 기존의 인터넷에 정적으로 연결되어 있는 컴퓨터 시스템 외에도 천만 단위의 사용자들이 무선 단말을 통하여 각종 서버 시스템에 접속하게 된다.
DMB, IPTV : 영상 컨텐츠로의 접근에 대한 자유도가 한 차원 높아짐에 따라 기반 서버 시스템에의 부하 또한 한 차원 높아진다.
BcN : 모든 디지털 정보 매체가 통일된 네트워크 시스템을 통하여 제공됨으로써 시너지 효과에 의한 기하급수적인 정보 유통의 증가가 예측되며, 이를 지원하기 위한 인프라 시스템의 고성능화가 필수적이다.

위와 같은 점들을 고려할 때, 현재보다 더욱 컴퓨팅 시스템에 대한 의존도가 높아지게 된다. 따라서 HDD의 응답속도 문제는 이러한 서버 시스템에서 더욱 크게 문제를 일으키게 되므로 이에 대한 대책이 필요하다.

플래시 메모리 자체는 산업계에서 이용되기 시작한지 10년이 넘었으나, 서버 시스템에서의 저장장치로 사용하기 위한 연구는 거의 이루어진 바 없다. 그러나 여러 가지 요소를 고려할 때, 서버 시스템용 저장장치로서도 충분히 사용할 수 있을 정도의 기반 기술들이 성숙하였다고 고려된다. 또한, 최근 들어 플래시 메모리의 가격이 낮아지고 있다. 그러므로 본 논문에서는 서버용 저장 시스템으로서의 플래시 메모리 저장 장치의 가능성과 그 설계 요건을 분석하고, 이에 기반하여 서버 시스템에 적합한 플래시 메모리 저장장치의 구조를 설계하였다.

2. 관련연구

현재 주로 사용되는 플래시 메모리에는 NOR형과 NAND형 등 두 가지가 있다. 기존에 많이 사용되던 NOR 플래시 메모리는 집적도가 낮고 고가이며 읽기 속도가 빠르지만 쓰기 속도가 느리기 때문에 운영체제나 응용프로그램 등의 코드 저장에 많이 사용되고 있다[1]. 반면, NAND 플래시 메모리는 단일 칩으로도 대용량이고 NOR 플래시 메모리에 비해 비용도 저렴하며 쓰기 속도가 빠르기 때문에 멀티미디어 데이터의 저장에 많이 사용되고 있다[2]. 따라서 NAND 플래시 메모리를 내장형 기기의 기본 저장장치로 사용하게 되면 기본 저장용량을 크게 할 수 있으며 기기의 비용도 낮출 수 있게 된다. 또한, 최근 출시되는 많은 수의 마이크로 컨트롤러들은 NAND 플래시 메모리를 제어할 수 있는 기능이 탑재되어 있어 NAND 플래시 메모리로부터 직접 부팅이 가능하므로, 기기들이 NOR 플래시 메모리와 같은 부가적인 부트용 저장장치 없이 NAND 플래시 메모리만으로 저장장치를 구성할 수 있다.

플래시 메모리는 기존의 HDD와 같은 다른 저장매체와 다른 몇 가지 제약사항들이 있다. 첫 번째로 한 번 데이터를 쓴 영역에 새로운 데이터를 쓰기 전에 반드시 그 영역을 지워야 한다는 점이다.

두 번째로 표 2에서 볼 수 있듯이 지우기 작업의 경우

다른 작업에 비하여 소요시간이 상당히 크기 때문에 이에 주의하여야 한다.

[표 2] 플래시 메모리의 작동 속도

Read	Write	Erase
130㎲/page	300㎲/page	2,000㎲/block

세 번째, 블록 별로 삭제 횟수가 제한(약 1,000,000회)되어 있기 때문에, 특정 블록에 대해 반복적으로 쓰기 연산이 수행되면 전체 플래시 메모리의 수명과 성능을 떨어뜨리게 된다.

위와 같은 제약을 감추고 플래시 메모리를 저장매체로 사용하기 위한 기법에는 파일시스템과 플래시 메모리 사이의 미들웨어인 플래시 변환계층 (Flash Translation Layer, FTL)을 사용하는 방법[3]과 플래시 메모리를 위한 전용 파일시스템을 사용하는 방법[4-8]이 있다. 플래시 변환계층은 플래시 메모리를 일반저장장치로 에뮬레이션 해줌으로써 상위에서 일반 파일 시스템이 동작할 수 있게 해준다. 하지만 플래시 변환계층을 사용하는 방법은 CPU 자원의 비효율적인 사용, 플래시 메모리 공간낭비, 주소변환에 필요한 메모리 사용량 증가, 특허문제로 인한 구현상 제약 등 여러 문제점이 있다. 이러한 문제점을 개선해 추가비용 없이 파일시스템에서 효율적으로 플래시 메모리를 제어하기 위해서 JFFS2 (Journaling Flash File System 2)와 같은 플래시 메모리 전용 파일시스템이 개발되었다[4]. 그러나 JFFS2는 마운트 성능과 읽기/쓰기 성능이 느린 문제가 있었다. 이를 보안하여 개발된 NAND 플래시 메모리 전용 파일 시스템이 YAFFS (Yet Another Flashing File system)[5]이다. 그러나 YAFFS는 JFFS2보다 안정성에서 성능이 떨어지는 문제점을 가지고 있다. 따라서, 최근 YAFFS를 기반으로 성능을 개선한 NFFiS (Nand Flash memory File System)[6], CFFS (Core Flash File System)[7], NAMU (NAnd flash MUltimedia file system)[8]등이 개발되었으며, 많은 연구가 이루어지고 있다. 그러나 위와 같은 연구들은 비교적 소용량의 임베디드(embedded) 시스템을 대상으로 한 파일 시스템으로서 서버 시스템을 위한 대용량 저장장치에 이용하기에는 무리가 많다.

3. 대용량 고성능 저장장치의 설계

3장에서는 앞절에서 설명한 플래시 메모리의 특성을 고려하면서 대용량 고성능 저장장치를 구성하기 위한 설계 요건을 설명하고, 이를 만족하는 구조 및 작동 메커니즘에 대하여 설명한다. 이후로 본 논문의 결과물인 저장장치의 명칭을 HiSFA(High Speed Flash memory Array)로 지칭하도록 하겠다.

3.1 설계요건

HiSFA를 설계함에 있어 기본적인 설계요건은 기존의 서버 시스템용 HDD 기반 저장장치의 일반적인 요구사항과 동일하다. 이는 HiSFA가 서버 시스템의 요구를 만족시켜야 하기 때문이다. 따라서 HiSFA가 추구하는 설계요건은 다음과 같다.

높은 IOPS(I/O operations Per Second): HDD의 기록밀도가 증가하면서 전송속도(transfer rate)는 100MB/sec에 [15.4K] 접근할 정도로 증가하였으나 접근시간(access time)은 별로 줄어들지 않았다. 다중 사용자의 요구를 동시에 처리하여야 하는 서버시스템의 경우, HDD 플래터(platter) 여기저기에 분산되어 있는 파일들에 접근하여야 하므로 전송속도보다 접근시간이 더욱 중요하다. 그러므로, 1초당 몇회의 I/O 요청을 처리하였는가라는 의미의 IOPS는 서버 시스템용 저장장치에 있어서는 매우 중요한 성능척도이다.

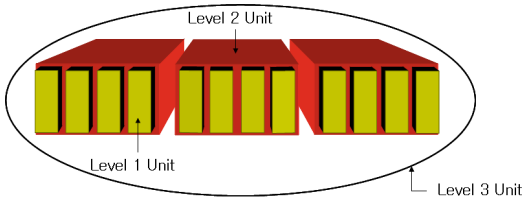
고장감내(falut-tolerance): 다수의 모듈로 이루어진 저장장치는 이를 구성하고 있는 모듈의 수에 비례하여 MTBF(Mean Time Between Failure)가 감소한다. 따라서 이에 대비하여 일부모듈에 장애가 발생하여도 전체적으로는 기능을 계속 수행하여야 한다. 특히, 플래시 메모리는 삭제 작업(erase operation)의 빈도에 따라 그 수명이 정해지므로 수명이 다한 플래시 메모리 모듈의 교환작업이 발생하게 되므로 고장감내 기능은 필수적이다.

유지보수의 용이성(Easy maintenance): 앞의 고장감내 기능과 연관되어 장애가 발생한 모듈의 교체 및 수리 등, 유지보수작업이 서버 시스템의 작동에 영향을 주지 않고 이루어져야 한다. 실제 저장장치 제품의 제작에 있어서는 이러한 유지보수의 용이성을 고려한 여러 가지 설계들이 상당수 기계적 설계에 반영, 즉 각 모듈의 크기, 형태, 탈착의 용이성 등 본 논문의 범위를 넘어서는 바 본 논문에서는 이러한 기계적 특성에 대해서는 언급하지 않고 기능적 특성에 대해서만 유지보수의 용이성을 고려하도록 한다.

3.2 HiSFA 구성 및 작동

앞 절에서 언급된 설계목표를 기반으로 하여 설계된

HiSFA는 3단계의 구조를 가진다. 그림 1에서 보는 바와 같이 최소단계인 Level 1 Unit(이하 L1U라 약칭)가 다수개(4~8개) 묶여 Level 2 Unit(이하 L2U라 약칭)를 구성하고 L2U가 다수개(2개 이상) 묶여 Level 3 Unit(이하 L3U라 약칭)를 구성하는데 이 L3U가 하나의 HiSFA이다. L1U, L2U는 모두 작동중에 개별적인 착탈이 가능하도록 구성된다. 즉, 특정 L1U 또는 L2U의 장애 발생 시에 이를 전체 HiSFA는 작동중인 상태로 교환 및 수리가 가능하도록 구성되는 바 이는 서버 시스템의 필수적 기능인 고장감내 기능을 지원하기 위해서이다. L1U, L2U 그리고 L3U의 구조 및 기능에 대하여 간략히 설명하도록 하겠다.



[그림 5] HiSFA 저장장치의 3단계 구조

3.2.1 Level 1 Unit(L1U)

가장 기본적 단위로서 시중에서 흔히 볼 수 있는 USB 플래시 메모리를 떠올리면 가장 유사하다고 볼 수 있다. 다수개의 L1U가 L2U에 연결되는 형식으로 장애시 즉시 교환 및 수리를 위해 핫스왑(hot-swap)기능을 가지고 있다. 하나 이상의 NAND 플래시 메모리 칩으로 구성되며 내부적으로 컨트롤러를 보유할 수도 있다. 컨트롤러가 L1U에 있는 경우 자체적으로 오류검사(ECC 등)를 수행할 수도 있다.

3.2.2 Level 2 Unit(L2U)

하드웨어적으로 볼 때 L2U는 다수의 L1U 및 이와 연결되는 접속부, L2U의 동작을 제어하는 제어부, 비상시에 내장된 자체 배터리로 작동를 지속할 수 있는 전원부 등으로 구성된다. 논리적으로 볼 때 L2U는 다수개의 L1U를 장애시 복구를 지원하는 메커니즘(RAID 1, RAID 5 등)으로 묶은 단위이다. 그러므로 L2U는 기존의 HDD RAID 시스템과 동치이다. 대다수의 서버 시스템용 HDD RAID 시스템이 그러하듯이 L2U도 개별 L1U의 장애시 이를 HiSFA 작동중에 제거할 수 있으며, 이 경우 장애복구 메커니즘(RAID 1, RAID 5 등)에 따라서 HiSFA의 성능이 일시 저하될 수 있다.

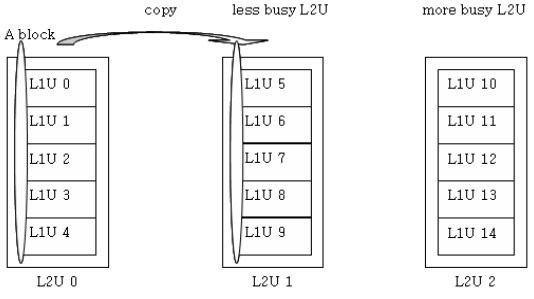
L2U의 제어부는 RAID 1, RAID 5 등의 메커니즘을 수행할 수 있어야 하므로 자체 CPU, 메모리, 캐시 등을 보유하고 있으며 L3U와의 연결을 위한 버스(Bus) 제어도 수행하여야 한다. 주목할 점은 L2U의 캐시는 쓰기(write) 전용 캐시로 동작을 한다는 점이다. 읽기(read) 작업용 캐시는 L3U에서 보유하며 제어한다.

3.2.3 Level 3 Unit(L3U)

L3U는 다수(2개 이상)의 L2U로 구성된 L2U 배열(L2U array)라 보면 된다. 즉, L2U를 하나의 HDD로 볼 때, L3U는 스트라이핑(striping)을 지원하는 HDD 배열(HDD array)과 유사하다. 차이점이 하나 존재하는데 HDD 배열의 스트라이핑은 하나의 논리적 블럭(block)을 전체 디스크에 분산시키나 L3U의 경우 하나의 논리적 블록은 하나의 L2U에 저장되는 바, 엄밀히 표현하자면 인터리빙(interleaving)적 요소가 표함되어 있다.

하드웨어적으로 볼 때, L3U는 다수의 L2U 및 이와 연결되는 버스, 동작을 제어하는 CPU 및 메모리, 읽기 전용 캐시 그리고 비상시를 위한 자체 전원 공급을 위한 배터리 등으로 구성된다.

L3U는 캐시에 상주한 블럭을 제외한 I/O 요구를 해당 L2U에 위임한다. 다만, 가비지 콜렉션(garbage collection, 이하 GC라 약칭) 작업만은 L3U에서 직접 제어한다. HiSFA의 GC는 기존의 플래시 파일 시스템에서 이루어지는 GC와는 상이한 과정을 통해 이루어지며 이를 시스톨릭 2 페이스 GC(Systolic 2-Phase GC, 이하 SGC라 약칭)라 일컫는다. 본 논문의 지면 관계상 HiSFA에서 쓰이는 모는 작업들의 알고리즘들을 본 논문에서 쓸 수 없으나 SGC의 경우 기존의 GC와 매우 상이한 관계로 SGC의 과정을 간략하게 아래에 기록한다. SGC의 대략적인 개념은 그림 2를 참고하면 쉽게 이해될 수 있을 것이다.



[그림 6] Systolic 2-Phase Garbage Collection

operation SGC_phase_1

1. select block to GC
2. select target L2U to move the chosen block
3. copy the block to the chosen L2U
4. adjust FTT(Flash Translation Tree)
5. Set delayed GC flag of the stripes (in FTT) and return

operation SGC_phase_2

1. search through FTT to find GC flag set block
2. select a block to erase
3. perform erase on each L1U alternately

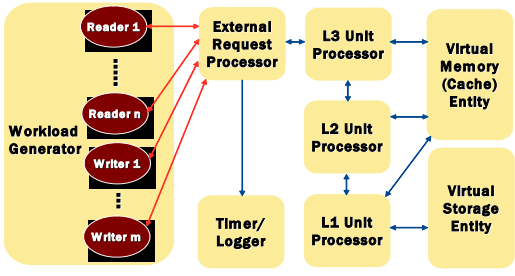
operation read_wihle_SGC_phase_2

1. read a block except locked L1U
2. compose whole block from parity block

4. 성능평가

4.1 시뮬레이터

본 논문에서는 첫째, HiSFA의 성능을 실험하고 간접적으로나마 검증하고, 둘째, HiSFA를 실제 제작할 경우 실행착오를 최소화하기 위한 각종 파라메터를 실험할 도구로서의 시뮬레이터를 제작하고 이를 통하여 시뮬레이션을 실행하였다. 본 논문의 지면 관계상 시뮬레이터의 세부적 설명은 생략하도록 하며 대략적인 구조는 아래의 그림 3과 같다.



[그림 7] HiSFA 시뮬레이터의 내부 구조

위의 시뮬레이터를 이용한 최초 시뮬레이션을 수행한 결과 실제 프로토타입을 통해 수행한 결과와 너무도 큰 성능 차이를 보이는 바, 이는 버스, 디바이스 드라이버(device driver) 그리고 운영체제의 오버헤드(overhead)가 추정치보다 크기 때문이었다. 이에 프로토타입의 결과를 분석하고 오버헤드 추정치를 수정하여 시뮬레이터를 수행한 결과 프로토타입의 성능 수행 결과와 유사한 결과가 도출되었다. 시뮬레이션의 결과와 프로토타입의 실험결과가 대체로 유사한 바, 지면 관계상 시뮬레이션의 결과는 생략하기로 한다.

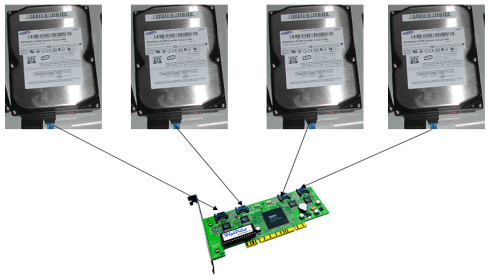
4.2 프로토타입

앞장에서 언급한 바와 같이 HiSFA를 실제로 제작하자면 다수의 컨트롤러 칩과 제어보드, 버스(Bus) 제어부 등 다양한 하드웨어 모듈을 제작하여야 하는 바 기간상으로는 1년 이상, 비용으로는 최소 1억 이상의 비용이 필요하다. 즉, HiSFA의 성능을 검증할 수 있는 프로토타입을 제작함에 있어 HiSFA의 기능 전체를 보유한 프로토타입의 제작은 본 논문의 범위에서는 불가능하므로 HiSFA의 기능 중 일부만을 지원하는 프로토타입을 최대한 단기간에, 최대한 저렴한 비용으로 제작할 수 있는 접근방법을 택하였다. 그 방법으로 플래시메모리 모듈은 기존의 USB 플래시메모리로 대체하고, 버스는 USB 허브로 대체하였으며 컨트롤러는 PC로 대체하였다(그림 4, 그림 5 참조). 기존의 USB 플래시 메모리를 채용함으로서 발생하는 제한점은 HiSFA의 여러 기능 중 읽기 기능(read operation)만을 실험할 수 있다는 것이다. 그러므로 본 절에서는 프로토타입의 성능을 실험함에 있어 읽기 기능만의 전송속도 및 응답시간을 실험하였다.

프로토타입의 성능을 기존 HDD 저장장치의 성능과 비교하기 위하여 4개의 HDD로 구성된 RAID 시스템(그림 6 참조)을 동일한 조건하에서 성능 평가하였다.

[그림 8] 4개의 L1U로 구성된 하나의 L2U

[그림 9] 4개의 L2U로 구성된 하나의 L3U



[그림 10] 4개의 SATA HDD로 구성된 RAID 시스템

4.3 프로토타입 성능 분석

앞절에서 언급한 바와 같이 성능 분석에는 표 3과 같은 구성의 HDD RAID와 HiSFA 프로토타입이 이용되었다.

[표 3] 성능 실험에 사용된 HiSFA 프로토타입과 HDD의 비교

	HiSFA	HDD RAID
매체	NAND Flash Memory	HDD
버스	USB	SATA
컨트롤러	PC(USB device driver, HiSFA read module)	SATA RAID controller card
매체 개수	8	4
단위 매체 용량	2GB	200GB

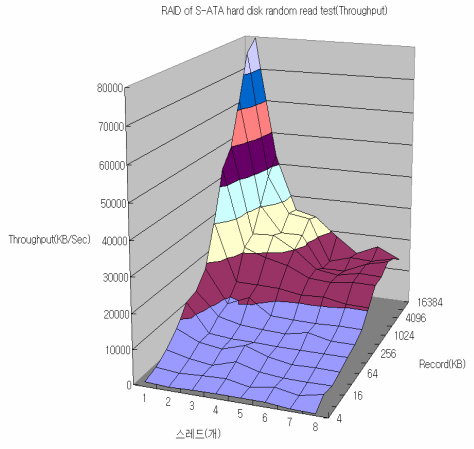
HiSFA와 HDD RAID 성능 실험 결과의 공통점은 한번에 읽어오는 레코드(record) 크기가 커질수록 전송속도(throughput)는 증가하고 IOPS는 줄어드는 것이다. 한 번에 읽어오는 레코드 크기가 커질수록 파일을 전송하는 시간외의 오버헤드가 줄어들기 때문에 전송속도는 증가하고, 한 번에 읽어오는 레코드 크기가 작으면 IO 시간이 줄어들기 때문에 IOPS는 증가한다.

실험 결과에서 스레드의 수는 서버에서 동시에 IO작업을 요구하는 프로세스의 수에 해당된다. HDD RAID의 경우 이 스레드가 2개 이상이 되면 급속도로 전송속도가 줄어든다. 그 이유는 데이터 파일들이 디스크 상에 물리적으로 멀리 떨어져 있기 때문에 동시에 파일을 읽을 때 디스크 헤더가 움직이는 시간 때문이다. 또한 그 이유 때문에 IOPS도 스레드의 개수에 상관없이 거의 일정하다.

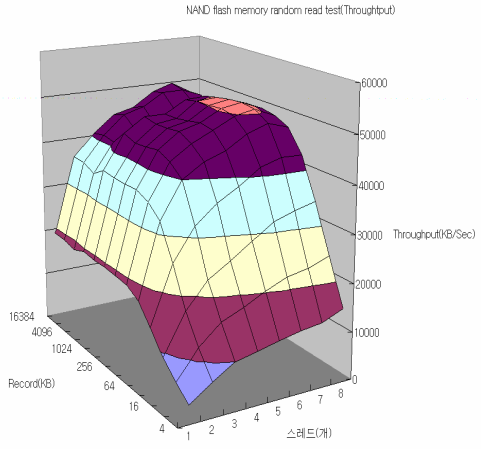
HiSFA의 경우 스레드가 늘어날수록 전송속도가 테스트 컴퓨터 시스템의 전송속도 한계까지 증가한다. 그리고 IOPS도 스레드가 늘어날수록 증가하는데, 특히 작은 크기의 레코드 단위로 읽을 때 IOPS는 많이 늘어난다. 이것은 HiSFA가 파일을 읽을 때 파일의 위치와 상관없이 파일을 읽는 속도가 일정하기 때문이다.

4.3.1 전송속도

전송속도 측면에서 스레드가 한 개일 때를 제외하고 HiSFA가 레코드의 크기가 클 때는 2배, 작을 때는 10배 이상의 성능을 보이며 HDD RAID를 성능우위를 보여 주었다. 그림 7의 전송속도(높이)를 보면 1개의 스레드가 16MB크기의 파일을 읽어들일 때 최대의 속도를 보이며 파일 크기가 줄어들수록, 스레드 개수가 늘어날수록 전송속도가 급격히 감소하고 있다. HiSFA 그림 8의 경우 스레드가 늘어날수록, 레코드의 크기가 클수록 전송속도가 증가하며 최대 50MB/s 부근에서 최대 전송속도이다.



[그림 11] HDD RAID throughput 그래프

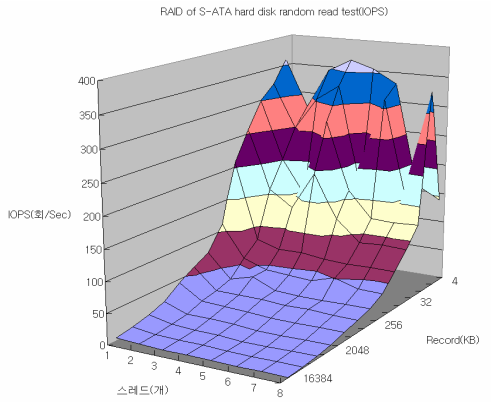


[그림 12] HiSFA throughput 그래프

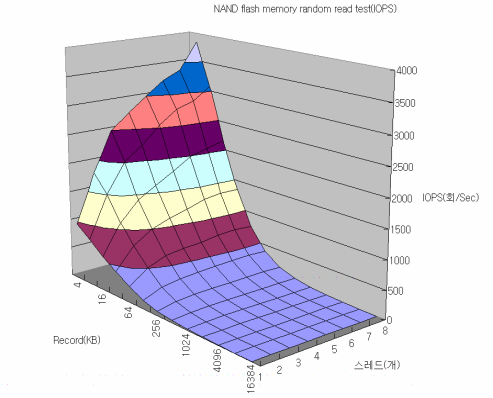
4.3.2 IOPS(I/O operations Per Second)

IOPS 측정결과를 보면 레코드의 크기가 100KB 이하일 때 HiSFA가 10배 이상의 성능으로 HDD RAID 성능을 뛰어

넘고 있다. 아래 그래프들, 그림 9, 그림 10중 좌측이 HDD RAID이며, 우측이 HiSFA이다. 주목할 것은 좌측그래프의 경우 IOPS최대치가 400인 반면 우측 그래프의 경우 IOPS최대치가 4,000인 점이다. HDD RAID의 경우 스레드의 개수에 관계없이 파일크기(레코드)가 증가할수록 IOPS가 급감하고 있다. HiSFA의 경우에도 파일(레코드)의 크기가 증가할수록 IOPS가 급감하고 있으나, 64KB 이하에서는 스레드 개수가 많을 수록 IOPS가 증가함을 알 수 있다. 특히 이 결과는 웹 서버의 저장장치로 HiSFA를 사용할 경우 기존의 하드 디스크를 웹 서버의 저장소로 사용한 것보다 훨씬 좋은 성능을 발휘할 것임을 예상할 수 있게 한다. 웹 서버가 클라이언트의 요청에 응답하기 위해 저장소로부터 읽어야 하는 파일의 크기는 수~수십 KB가 대부분이기 때문이다.



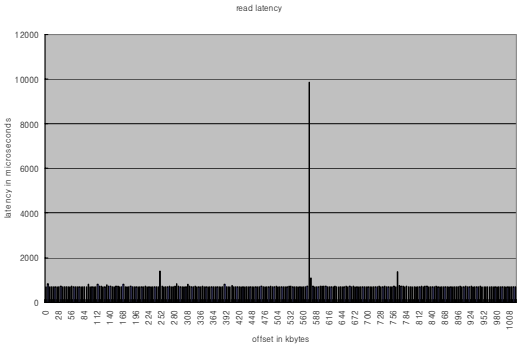
[그림 13] HDD RAID IOPS 그래프



[그림 14] HiSFA RAID IOPS 그래프

4.3.3 응답시간

응답시간의 경우 IOPS에 이미 응답시간이 포함되어 있으므로 별도의 분석이 큰 의미를 가지지 못하나 본 논문의 정량적 목표중 하나가 '평균 2ms이내의 응답시간'을 성취하는 것이므로 응답시간 실험을 수행하였으며 그 결과는 아래의 그래프와 같다. 거의 모든 경우 응답시간이 1ms 이하임을 알 수 있다.



[그림 15] HiSFA의 응답시간 그래프

5. 결론

본 논문에서는 네트워크 시대, 정보화시대에 급증하는 각종 서버시스템을 위하여, 기존의 HDD 문제점을 보완할 수 있는 새로운 고성능 저장장치를 NAND 플래시 메모리를 이용하여 제작할 수 있는 기술 개발에 그 목표를 두었다. 이에 HiSFA가 설계되었으며, 이 설계를 기반으로 하여 그 성능을 예측할 수 있도록 시뮬레이터를 개발하였다. 결과물의 검증에 있어 실제로 HiSFA를 제작하여 실험하는 것이 가장 타당한 검증 방법이겠으나 제한적인 여건상 본 논문에서는 설계된 저장장치의 기능 중 일부만 수행 가능한 프로토타입을 제작하였으며 그 성능을 검증하였다. 그 결과 기존 HDD 기반 RAID 시스템에 비하여 최대 10배 정도의 성능을 보임을 확인할 수 있었다.

본 논문 결과물의 문제점으로는 저장공간당 비용이 기존의 HDD에 비하여 상당히 높다는 점을 들 수 있으나 현재 NAND 플래시 메모리 기술의 빠른 발전에 힘입어 저장공간당 비용이 급격하게 낮아지고 있다는 점과 IOPS당 비용에서는 거의 대등한 수준이라는 점은 본 논문 결과물의 활용에 있어 매우 긍정적이라 할 수 있겠다.

참 고 문 헌

[1] Intel Corp., "3 Volt Synchronous Intel Strata Flash Memory," http://www.intel.com/.

[2] Samsung Electronics Co., “NAND Flash Memory,” http://www.sec.co.kr/.
[3] Intel Corporation, “Understanding the flash translation layer (FTL) specification”, Application Note 648, 1998.
[4] D. Woodhouse, Red Hat, Inc., “JFFS : The Journaling Flash File System,” http://linux-mtd.infradead.org/~dwmw2/jffs2.pdf/.
[5] Aleph One Company, “YAFFS : Yet Another Flash Filing System,” http://www.yaffs.net/.
[6] Sang-Oh Park, Sung-Jo Kim, “A Fast Mount and Stability Scheme for a NAND Flash Memory-based File System”, Journal of KIISE : Computer System and Theory, Vol.34, No.12, December 2007.
[7] Seung-Ho Lim, Kyu-HoPark, “An Effective NAND Flash File System for Flash Memory Storage”, IEEE Transactions on Computers, Vol.55, No.7, JULY 2006.
[8] Sang-Oh Park, Sung-Jo Kim, “An Efficient Multimedia File System for NAND Flash Memory Storage”, IEEE Transactions on Consumer Electronics, Vol.55, No.1, FEBRUARY 2009.

류 동 우(Dong-Woo, Ryu) **[정회원]**

- 2004년 2월 : 중앙대학교 정보대학원 컴퓨터공학과(공학석사)
- 2007년 8월 : 중앙대학교 일반대학원 컴퓨터공학부(박사수료)
- 1995년 5월 ~ 현재 : 가톨릭대학교 중앙의료원 정보지원팀

<관심분야>
모바일, 정보보안, 소프트공학, U-Health, 유비쿼터스

김 상 욱(Sang-Wook, Kim) **[정회원]**

- 1994년 2월 : 중앙대학교 대학원 컴퓨터공학과(공학석사)
- 2000년 8월 : 중앙대학교 일반대학원 컴퓨터공학부(박사수료)
- 2007년 3월 ~ 현재 : 이너비트(주) 책임연구원

<관심분야>
이동컴퓨팅, 임베디드 소프트웨어, 임베디드 DBMS

맹 두 열(Doo-lyel, Maeng) **[정회원]**

- 2004년 2월 : 중앙대학교 정보대학원 컴퓨터공학과(공학석사)
- 2008년 8월 : 중앙대학교 일반대학원 컴퓨터공학부(박사수료)
- 2001년 1월 ~ 현재 : 한국정보보호진흥원 선임연구원

<관심분야>
정보보안, 임베디드 소프트웨어, 유비쿼터스컴퓨팅