리눅스 기반의 SSD 상에서 동작하는 파일 시스템의 I/O 분석 모듈 설계

김소연, 박치현, 노홍찬, 박상현
연세대학교 컴퓨터학과
e-mail : {sykim, tianell, fallsmal,sanghyun}@cs.yonsei.ac.kr

A Study on I/O performance analysis Architectures for file system based on SSD

So-Yeon Kim,Chi-Hyun Park,Hong-Chan Roh,Sang-Hyun Park
Dept. of Computer Science, Yonsei University

요 약

SSD 는 하드 디스크와 다른 구조를 갖고 있으며, 단편화된 잦은 쓰기 연산에 취약하다는 점 등 기존의 환경과는 차이가 있기 때문에 이런 환경에서 발생하는 연산, 특히 I/O 연산에 대한 분석이 필수적이다. 기존의 I/O 연산 측정 도구로 사용되던 벤치마크를 이용하여 SSD 의 성능을 측정할 경우에는 상위단계에서의 읽기, 쓰기 성능만 분석되기 때문에 하위단계에서 실제로 SSD 상의 I/O 연산의 수행 성능을 정확히 측정하기 어렵다. SSD 는 내부 저장 알고리즘의 효율성에 따라서 성능 차이가 크기 때문에 정확한 성능을 측정할 수 있는 분석 도구가 필요하다. 본 논문에서는 SSD 상에서의 I/O 연산의 계층적 분석을 위한 모듈을 제안한다.

1. 서론

SSD(Solid State Disk) 는 비휘발성의 메모리 반도체인 다수의 플래시 메모리 칩들을 포함하는 저장장치로서 일반적으로 해당 플래시 메모리 칩들의 데이터 공간을 효율적으로 관리하기 위한 플래시 변환 계층(Flash Memory Translation Layer), CPU, RAM 등을 자체적으로 내장하고 있다. 이러한 이유로 SSD 는 플래시 메모리의 우수한 특성인 소비전력이 적고 외부 충격에 강하며 저장장치에 대한 접근 비용이 매우 작은 특성을 가진다. 하드디스크에 비해 매우 안정적이고 기존 컴퓨터 시스템에서 가장 큰 병목 현상이었던 저장장치에 대한 접근시간을 줄일 수 있는 특성들로 인해 최근 SSD 가 하드디스크를 대체할만한 차세대 저장장치로서 주목 받고 있다[1][2].

SSD 구조의 문제점은 다수의 채널에 의해 데이터가 전송되고 인터리빙에 의해 한 단위로 쓰여지는 다수의 블록이 하나의 논리적인 슈퍼 블록으로 구성된다는 것이다. 내부 공간 갱신이 불가능한 플래시 메모리의 특성 때문에 매우 작은 공간의 쓰기 요청에도 최대 2MB 까지 구성되는 슈퍼 블록 전체가 다시 다른 여유 공간에 쓰여지는 비효율을 초래하게 된다. 이러한 쓰기 연산에 의해 연속되지 않은 공간에 대한 파편화된 쓰기 요청이 많아지는 경우 SSD 는 각 블록의 지우기 횟수를 빠르게 소모하게 되고 결국 몇 개의 블록의 지우기 횟수가 역치를 초과해 저장장치로서의 수명을 단축시키게 된다. 또한 쓰기 성능에 있어서도 연속된 공간에 대한 대용량 쓰기 요청은 다채널 인터리빙의 효과를 보게 되어 플래시 메모리에 단위 공간당 지연 시간이 줄어드는 효과를 볼 수 있다. 하지만 임의의 공간에 대한 작은 용량의 쓰기 요청의 경우 같은 슈퍼 블록에 속한 데이터들에 대해 불필요한 쓰기 연산이 수행되어 다채널 인터리빙에 의해 쓰기 성능이 개선되는 효과를 보지 못하게 된다.

위에서 언급한 바와 같이 SSD 는 단편화된 잦은 쓰기 연산에 대해 취약한 문제점을 가지고 있다. 이러한 문제점을 하드웨어 계층에서 해결 하려는 노력이 진행 중에 있지만 그 상위에서 동작하는 핵심 소프트웨어들에서도 이러한 단편화된 쓰기 회수를 줄이려는 노력이 필요하다. 이렇게 하기 위해서 상위 소프트웨어들이 어떤 I/O 작업을 발생시키는 지에 대한 분석을 통한 개선이 요구된다.

본 논문에서는 VFS 내에서 SSD 파일 시스템에 대한 I/O 분석 모듈을 설계한다. 이를 통해 SSD 상의 인덱스 구조를 설계하거나 실제 SSD 의 성능을 사용자 혹은 응용프로그램이 측정할 때 정확한 수치를 제공해 줄 수 있게 한다.

본 논문의 구성은 다음과 같다. 2 장의 관련연구에서는 기존의 파일시스템 I/O 분석 모듈과 페이지 크기에 따른 B+트리의 성능 평가에 대해 알아보고 3 장에서는 본 논문에서의 I/O 분석 모듈 설계에 대해 설명한다. 마지막으로 4 장에서 결론을 맺으며 향후 과제

를 기술한다.

2. 관련연구

2.1 기존의 파일 시스템을 위한 I/O 분석 모듈

파일 시스템 측정 도구인 IOzone 을 이용하여 리눅스에 탑재된 플래시 메모리용 파일 시스템인 JFFS2[3]을 테스트하여 읽기, 쓰기를 측정한다.

표 1. IOzone 을 인용한 jffs 를 테스트한 결과

./iozone -Razb jffs2.wks -g 1M -i 0 -q 2k			
The top row is records sizes, the left column is file sizes			
Write Report			
	0	1	2
64	233	386	531
128	226	374	528
256	163	376	559
512	210	371	558
1024	229	373	554
Re-writer Report			
	0	1	2
64	291	554	974
128	291	540	974
256	290	547	995
512	290	545	981
1024	289	545	980

표 1 처럼 IOzone 은 페이지 크기별로 각 파일들에 대한 읽기, 쓰기 등의 대역폭을 보여준다. 상기 벤치마크를 이용하여 SSD 의 성능을 측정할 경우에는 상위단계에서의 읽기, 쓰기 성능만 분석되기 때문에 실제 SSD 상의 I/O 연산의 수행 성능을 정확히 측정하기 어렵다[4].

2.2 NAND 플래시 메모리에서 페이지에 따른 B+트리의 성능 평가

NAND 플래시 메모리를 저장 장치로 사용하는 데이터베이스 시스템에서 효율적인 인덱스 지원을 위하여 B+트리의 페이지 크기에 대한 플래시 메모리에서의 성능을 평가한다. B+트리의 페이지 크기에 따른 플래시 변환 계층 알고리즘의 성능을 비교, 분석한다.

플래시 변환 계층 알고리즘은 매핑 방법이나 물리적인 위치에 쓰는 방식에 따라 FMAX, BAST, FAST, MTBS 방법으로 분류할 수 있다. 각 알고리즘의 특성은 다음과 같다[5].

표 2. 알고리즘의 특성 비교

	FMAX	BAST	FAST	MTBS
데이터	고정섹터방식	고정섹터방식	고정섹터방식	고정섹터방식
맵핑	블록 매핑	혼합 매핑	혼합 매핑	블록 매핑
덮어쓰기	고정섹터방식→변동섹터방식	변동 섹터 방식	변동섹터방식	변동섹터방식
특징	각 블록당 1:1 로 복사블록을 사용	한 블록에 로그블록하나를 할당	로그블록을 순차쓰기용과 임의쓰기용으로 구분	블록내의 여유영역을 사용

표 3. 알고리즘별 실험결과(단위: 횟수) 1

알고리즘	연산	페이지 크기		
		512B	4K	16K
FMAX	쓰기	120,201	934,035	114,808
	소거	3,514	28,092	3,640
BAST	쓰기	449,280	1,068,925	143,214
	소거	36,649	36,742	3,490
FAST	쓰기	281,214	1,325,746	62,548
	소거	7,099	33,087	1,929
MTBS	쓰기	111,697	648,133	109,949
	소거	3,409	20,218	3,647

표 3 처럼 B+트리의 페이지 크기를 플래시 메모리의 소거 단위인 16K 로 했을 때 가장 좋은 성능을 보인다. B+트리에서 페이지 크기에 따른 각 알고리즘을 실험한 결과 읽기, 쓰기 단위인 섹터 혹은 섹터의 배수 단위로 하는 것 보다 소거 단위인 블록으로 B+트리의 페이지 크기를 결정하는 것이 더 좋은 성능을 보인다[6]. 이처럼 SSD 상에서 최적의 B+트리 성능을 나타낼 수 있는 알고리즘을 만들기 위해 선행되어야 할 중요한 작업 중 하나가 실제로 개선된 알고리즘의 정확한 성능의 측정이다. 기존 벤치마크들은 응용프로그램 단계에서 개선된 알고리즘의 성능을 측정하곤 했지만, 본 논문에서와 같이 VFS 단계에서 실제로 시스템 내부에서 I/O 연산을 처리하는 단위로 성능을 측정한다면 정확한 성능 결과를 얻을 수 있다.

3. 커널 환경에서 SSD 의 I/O 분석을 위한 모듈 설계

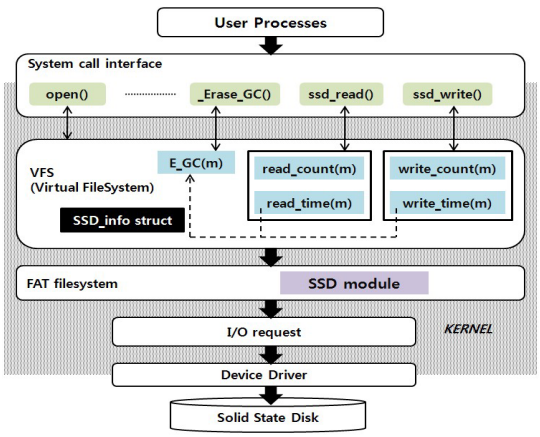

그림 1. 커널 환경에서 SSD 성능을 측정하기 위한 분석 모듈 구조

그림 1 은 리눅스 기반의 SSD 상에서 동작하는 파일 시스템의 I/O 분석 모듈 구조도 이다. 기존의 벤치마크는 응용(application)단계에서 I/O 이벤트의 발생 시간과 횟수를 측정하지만 본 논문에서 제안하는 성능 측정 모듈은 OS 의 파일시스템 내에 모듈 형태로 삽입되어 사용되도록 설계한다.

VFS 내에서 SSD 상에서 동작하는 파일시스템의 I/O 작업에 대해 횟수와 지연 시간 측정 결과 및 관련 통계 정보를 저장할 수 있는 메인 메모리 기반 데이터 구조를 설계한다. 그림 1 에 나와있듯이 SSD_info 구조체는 SSD 에 대한 읽기, 쓰기 횟수 및 지연 시간을 기록할 수 있는 VFS 내에 동적 메모리 공간이다. 정적인 메인 메모리 공간으로 설계할 경우 어플리케이션의 I/O 요청이 최대일 때를 대비하여 메인 메모리 공간을 할당하므로 많은 메인 메모리 공간을 낭비한다. 그러므로 I/O 연산의 요청 수에 따라 메인 메모리 공간을 동적으로 할당하고 또한 메모리 공간의 할당을 해제할 수 있는 매커니즘 및 효율적인 데이터 구조를 설계 및 구현한다. 데이터 구조는 아래 그림 2 와 같다. SSD_info 구조체는 각 프로세스 별로 가지고 있으며 같은 프로세스더라도 다른 파일들을 대상으로 성능을 측정한다면 여러 구조체를 가지고 있을 수 있다. 이럴 때 여러 구조체도 리스트로 연결되어 있다.

Struct ssd_info	
변수명	역할
PIDlist_node	PID 리스트 연결
PID	프로세스 식별
filename	I/O 대상 파일 이름
write_count	SSD 에 쓰기연산 수행 횟수
write_time	SSD 에 쓰기연산 수행에 걸린 시간
read_count	SSD 에 읽기연산 수행 횟수
read_time	SSD 에 읽기연산 수행에 걸린 시간
time	현재 성능을 측정하는 시간
next_pointer	다음 ssd_info 구조체를 가리키는 포인터
E_GC	erase 연산을 위한 가비지 컬랙션 및 웨어 레벨링 작업의 탐지

그림 2. SSD_info 구조체

SSD 에서 발생하는 읽기, 쓰기 횟수를 측정하는 모듈을 설계하기 위해 VFS 내부에 성능 측정을 수행할 수 있는 내부 함수를 설계한다. read_count(), write_count()는 I/O 작업에 대한 횟수 측정을 위해 각 어플리케이션이 요청하는 파일 읽기 또는 파일 쓰기 작업에 대해 SSD 상에서 파일시스템의 I/O 를 최소 I/O 단위로 파일 읽기 및 쓰기 횟수로 환산하여 이를 측정할 수 있도록 설계한다. 이를 위해 VFS 에서 파일 I/O 연산을 수행할 때 내부적으로 호출되는 하위 함수를 분석하여 실제 SSD 상에 동작하는 파일 시스템의 I/O 연산이 되는 단위를 고려하여 정확한 I/O 횟수를 구한다.

또한 VFS 내에서 SSD 상에 동작하는 파일 시스템의 읽기, 쓰기 지연 시간을 측정할 수 있는 모듈을 설계한다. read_time(), write_time()는 단위 I/O 작업에 대한 지연 시간 측정을 위해 각 어플리케이션이 요청한 파일 읽기 또는 파일 쓰기 작업에 대해서 파일 시스템의 I/O 를 SSD 상의 최소 I/O 단위로 환산하고 최소 I/O 단위에 대한 단위 파일 읽기 및 쓰기 시의 지연시간을 할당된 메인 메모리 공간 내에 기록한다. VFS 의 하위 I/O 연산 함수를 분석하여 실제 SSD 가 수행하는 최소 단위의 I/O 연산이 수행되면서 걸리는 시간을 측정한다.

VFS 상에서의 이러한 함수는 기존의 sys_write, sys_read 같은 시스템콜이 내부에서 불렸을 경우 이벤트를 발생시켜 실제 내부에서 발생하는 I/O 의 정확한 카운트를 측정한다. 실제 마운트되는 파일시스템단계에 이러한 모듈구조를 만들지 않은 이유는 마운트된 파일시스템에 종속되지 않게 SSD 상에서의 I/O 성능을 측정하기 위함이고, 실제 커널 내부에서 불리는 I/O 연산들은 마운트된 파일시스템에 고유하게 정의된 write, read 함수를 부르지만 마지막 단계에서는 파일시스템에서 공통적으로 사용하는 함수를 이용하여 저장장치에 I/O 를 하기 때문에 이러한 설계를 하였다. 또한 전체적인 구조에 있어서 VFS 내에서 읽기, 쓰기 및 시간 측정을 각 프로세스 별도로 독립적으로 측정할 수 있도록 설계한다. I/O 횟수 및 단위 I/O 작업 지연 시간 측정을 하던 모듈에 대해 해당 성능 분석이 요청된 프로그램에 대해서만 추적이 가능하도록 분석을 요하는 프로세스의 식별자를 등록 및 해제할 수 있는 시스템 콜을 추가한다. 커널에서 해당 시스템 콜들로부터 등록된 프로세스들에 대해서만 성능 분석이 수행될 수 있도록 프로세스 식별자를 빠르게 비교하고 확인할 수 있도록 한다.

VFS 내에서 SSD 상에서 동작하는 파일시스템의 I/O erase 작업에 대한 모듈을 설계한다. SSD 에 대한 읽기, 쓰기 지연 시간들을 각 요청의 쓰기 및 읽기 SSD 별로 군집화시키고 아웃라이어를 걸러냄으로써 가비지 컬랙션 및 웨어 레벨링을 탐지한다.

4. 결론

소비전력이 적고 외부 충격에 강하며 저장장치에 대한 접근 비용이 작으며 접근 시간을 적다는 특성을 가진다. 이런 특성들로 인해 최근 SSD 가 하드디스크를 대체할만한 차세대 저장장치로서 주목 받고 있다. 이런 특성화된 저장장치를 위한 I/O 분석 모듈이 개발되어야 한다. 본 논문에서는 SSD 을 위한 VFS 내에서 I/O 모듈 설계를 제안함으로써 SSD 을 위해 만들어진 파일 시스템들에 대해 기존의 분석 모듈보다 정확한 시간과 횟수 측정이 가능할 수 있는 방안을 제시하였다.

SSD 는 그 특성상 erase 연산이 요구된다. erase 연산이 플래시메모리의 성능에 큰 영향을 준다. 따라서 향후 VFS 상에서의 erase 연산에 대한 추가적인 연구가 필요하다. 또한 제안 분석 모듈의 SSD 에 대한 I/O 모듈 구조의 신뢰성까지 측정할 수 있는 도구 개발을 위한 연구가 필요하다.

참고문헌

[1]배영현 “고성능 플래시 메모리 SSD(Solid State Disk)설계 기술”, 한국정보과학회 논문집, 제 25 권, 2007 년, 6 월.

[2]Samsung Electronics,Co. “Solid State Disk Development Data Sheets”, http://www.samsungelectronics.com/, 2008.

[3]D. Woodhouse, “JFFS: The Journaling Flash File System”, In Proceedings of the Ottawa Linux Symposium(OLS), RedHat Inc., 2001.

[4]박상오, 김준오 “리눅스 기반의 NAND 플래시 메모리 파일 시스템에 대한 성능 측정 도구 설계”, 한국정보처리학회 추계학술발표회 논문집, 제 32 권, 2005 년, 11 월.

[5] 박원주, 유현석, 박성환, 김도윤, 박상원. “윈도우즈 파일 시스템에서 플래시 메모리의 FTL 알고리즘 성능 분석” 한국정보과학회 논문집, 제 23 권, 2005년, 7월.

[6]유현석, 전한별, 김도윤, 박상원 “NAND 플래시 메모리에서 페이지 크기에 따른 B+트리의 성능 평가”, 한국컴퓨터종합학술대회 논문집, 제 33 권, 2006 년, 6 월.