

임베디드 오디오 파일시스템의 클러스터 크기 연구

임창규, 이인환

한양대학교 전자통신컴퓨터공학부

e-mail : cklim@csl.hanyang.ac.kr, ihlee@hanyang.ac.kr

Cluster Sizing for Embedded Audio File Systems

Changkyu Lim, Inhwan Lee

Dept. of Electronics and Computer Engineering, Hanyang University

요 약

본 논문에서는 현실 세계에서 사용되고 있는 디지털 오디오 파일들을 사용하여 전통적인 파일시스템에서의 클러스터 크기들이 파일시스템의 입출력 처리시간 및 공간 사용률 등의 성능 요소에 미치는 영향들을 분석하고 이를 디지털 오디오 파일의 특성에 맞게 개선함으로써, 임베디드 오디오 파일시스템의 클러스터 크기에 관한 개선된 디자인을 제시하고자 한다.

1. 서론

전세계 시장조사기관인 In-stat/MDR 의 2005 년 6 월 보고서에 따르면 전 세계 디지털 오디오 디바이스의 수요는 오는 2009 년까지 1 억 4 백만 대에 달해 현재 보다 4 배 가까이 증가할 것이고, 2004 년의 매출규모도 약 45 억 달러로 2003 년에 비해 거의 200% 성장했다고 발표했다. 또한 이 보고서는 2005 년에 판매된 전체 MP3 플레이어 중에서 1 기가바이트 이상 용량의 디바이스가 차지하는 비중이 20%를 넘어설 것이며, 향후 그 저장용량 또한 급격히 커질 것으로 전망하고 있다 [1].

디지털 오디오 파일의 경우 보통 3~6 메가 바이트의 크기를 가진다. 이는, 평균 4 메가바이트의 파일 크기를 가정할 경우 4 기가바이트의 저장용량을 가진 임베디드 오디오 디바이스에는 약 1,000 개 정도의 오디오 파일을 저장할 수 있게 된다. 대부분의 경우 임베디드 오디오 시스템은 일반적인 컴퓨터 시스템보다 매우 낮은 성능을 가지므로 이에 대한 파일시스템의 효율성은 저장 매체의 고용량화가 진행될수록 더욱 중요한 요인이 되고 있다.

임베디드 오디오 시스템에 내장되는 저장매체는 컴퓨터 시스템에서 사용되는 전통적인 파일시스템에 의

해서 관리된다. 그러나, 이들 전통적인 파일시스템들은 저장되는 파일들의 크기가 대부분 10 킬로바이트 이하이고, 주로 임의 위치의 접근방식을 통한 텍스트 파일의 처리라는 과거의 컴퓨터 시스템상의 특성을 반영하여 설계됨으로써 [2], 현대의 임베디드 오디오 시스템의 특성을 충분히 반영하지 못하고 있다.

또한, 최근에는 멀티미디어의 특성을 고려한 멀티미디어 전용의 파일시스템들이 많이 연구되어 발표되고 있으나, 이러한 멀티미디어 파일시스템들은 대부분 디지털 비디오의 특성에 그 초점이 맞춰져 있어서 [3, 4], 이를 임베디드 오디오 시스템에 적용하기에도 비효율적이다.

따라서 본 논문에서는 현실 세계에서 사용되고 있는 디지털 오디오 파일들에 대해서, 전통적인 파일시스템에서의 클러스터 크기들이 이러한 파일들에 대한 입출력 처리시간 및 공간 사용률 등 파일시스템의 성능 요소에 미치는 영향을 평가해봄으로써, 임베디드 오디오 파일시스템을 디자인할 때의 이슈 중 하나인 클러스터 크기에 관한 개선된 디자인을 제시하고자 한다.

2. 임베디드 오디오 파일시스템의 클러스터링

디지털 오디오 디바이스는 내장되는 저장매체의 타입에 따라 하드디스크 타입과 플래시 메모리 타입으로 나눌 수 있다. 컴퓨터 시스템과의 호환성과 개발의 용이성으로 인해 하드디스크 타입의 경우 대부분 FAT (File Allocation Table) 파일시스템을 사용하고, 플래시 메모리 타입의 경우 자체적인 플래시 파일시스템 혹은 FTL (Flash Translation Layer)를 FAT의 하부 서비스 층으로 둔 FAT+FTL 형태의 파일시스템을 구성하고 있다.

2.1 하드디스크 시스템

하드디스크 상에서 파일시스템이 접근할 수 있는 가장 작은 공간의 단위는 섹터이며, 일반적으로 512 바이트의 크기를 가진다. 이 섹터는 파일을 디스크 상에 저장할 때, HPFS (High Performance File System)와 같이 파일이 필요로 하는 수만개의 개별적인 섹터로 할당하기도 하지만 대부분의 파일시스템은 섹터들을 연속된 섹터들로 이루어진 보다 더 큰 블록단위로 할당한다 [5]. 따라서 개별적으로 관리되는 512 바이트 크기의 섹터를 사용하는 4 기가바이트 디스크 볼륨의 경우 800 만개 이상의 개별적인 섹터들에 대한 정보관리가 필요해진다.

클러스터는 파일시스템에 의해 하나의 파일로 할당될 수 있는 가장 작은 디스크 공간의 단위가 되며, 이러한 섹터들의 배열 및 연결되어 있는 파티션이나 볼륨의 크기 등은 파일시스템의 성능과 디스크 이용률 등에 있어서 중요한 영향을 미친다 [6].

2.1.1 FAT 파일시스템에서의 공간 이용률

임베디드 오디오 파일시스템에 있어서 저장매체에 대한 공간 이용률은 중요한 성능 평가 요소가 된다.

일반적으로 클러스터의 크기가 클수록 파일의 메타데이터에 관한 오버헤드가 줄어들고, 연속적인 데이터 처리를 가능하게 함으로써 그 성능은 향상되나, 파일들이 정수 개의 클러스터들을 채울 가능성이 더 적어지기 때문에 저장 매체의 공간낭비가 커지게 된다 [6]. FAT 파일시스템에서 파일이 사용하고 있는 모든 클러스터는 FAT의 엔트리 속에 파일이 사용하고 있는 다음 클러스터와 연결시키는 클러스터 번호를 가지며, 그 클러스터 또한 자기 뒤에 오는 클러스터의 번호를 가짐으로써 하나의 파일에 대한 완전한 체인이 형성된다. 또한, 파일의 디렉토리 엔트리와 FAT에 있는 클러스터 엔트리의 결합을 이용함으로써 파일의 임의 위치에 대한 접근이 이루어진다.

디스크 타입의 임베디드 시스템에 적용된 대부분의 파일시스템은 FAT16 혹은 FAT32이며, 이 FAT 파일시스템에서의 하드디스크 파티션은 그림 2.1 과 같이 부트 섹터영역, FAT 섹터영역, Root 섹터영역 및 데이터 섹터영역의 구조로 되어있다.

시작주소 (LBA)	0	1	nFAT+1	2*nFAT+1	2*nFAT+1+nRootSize = nStartLBA	
섹터수	1	nFAT	nFAT	nRootSize	TCS	
영역	PT BR	FAT	FAT'	nSet-Root	RS	Data Sectors
	Boot		FAT	Root		Data

PT : Partition Table, BR : Boot Record
LBA : Logical Block Address FAT' : Copy of FAT

(그림 2.1) FAT의 파티션

또한 이러한 FAT 파일시스템 구조에서 각 영역의 크기는 다음과 같이 계산할 수 있다 [3].

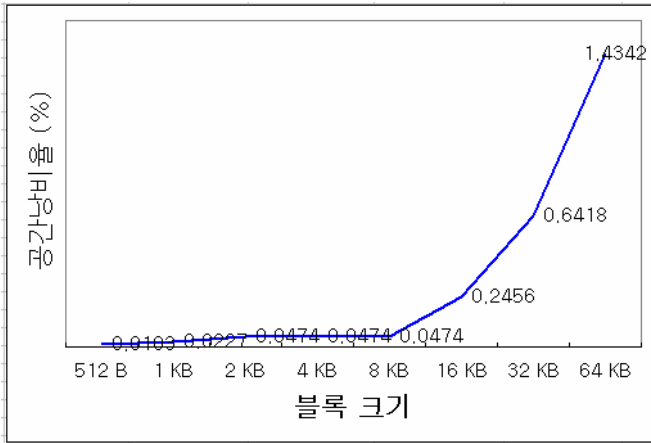
- ① 파티션의 총 클러스터 수
 $TC = RoundOff \{ nSec * (TS - 1 - nSetRoot) / (nSec * SC + 2 * nByte) \}$
 nSec : 섹터의 바이트 크기
 TS : HDD의 총 섹터 수
 SC : 클러스터당 섹터 수
 nByte : FAT의 바이트 수
- ② 파티션의 총 섹터 수
 $TCS = TC * SC$
- ③ 루트 디렉토리의 섹터 수
 $nSetRoot = RoundUp \{ (nRoot * 64) / (nSec * SC) \} * SC$
 nRoot : (nRoot*64)/nSec 를 정수값으로 하는 최소 값의 루트 파일 수
- ④ 섹터당 FAT의 크기
 $nFAT = RoundUp \{ TC * nByte / nSec \}$
- ⑤ 남은 섹터의 수
 $RS = TS - TCS - 2 * nFAT - 1 - nSetRoot$
- ⑥ 루트 디렉토리의 섹터 크기
 $nRootSize = TS - TCS - 2 * nFAT - 1 = nSetRoot + RS$
- ⑦ 루트 디렉토리의 총 파일 수
 $nRootFiles = RoundOff \{ nSec * nRootSize / 64 \}$
- ⑧ 데이터 영역의 시작 주소
 $nStartLBA = 2 * nFAT + 1 + nRootSize$
- ⑨ 데이터 영역의 끝 주소
 $nEndLBA = TS - 1$

임베디드 오디오 시스템에서 사용되는 현실세계의 파일들은 보통 3~6 메가바이트의 크기를 가진다.

2 기가바이트의 볼륨크기를 가지는 파일시스템에서 평균 4,134,998 바이트의 크기를 가지는 508 개의 디지털 오디오 파일들이 차지하는 공간은 2,100,578,984 바이트의 크기이어야 하나, 실제로 디스크 시스템에서 차지하는 물리적인 공간은 클러스터의 크기와 파일의 조각율에 따라 달라진다.

Cluster Size	512 B	1 KB	2 KB	4 KB	8 KB	16 KB	32 KB	64 KB
Actual Size (MB)	2.003	2.004	2.004	2.004	2.004	2.008	2.016	2.032

(그림 2.2) 디스크 공간 사용률

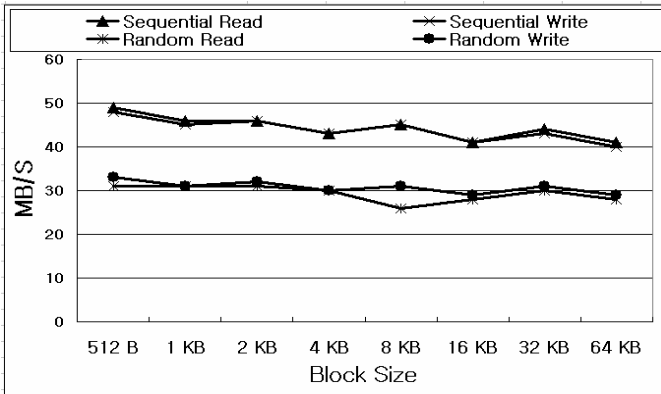


(그림 2.3) 디스크 공간 낭비율

따라서, 이러한 파일들에 대해서 파일시스템의 파일 조각율(Fragmentation Rate)을 최소화시킨 후 측정할 디스크의 공간낭비율은 그림 2.3 에 나타난 바와 같이 일반적인 파일시스템의 경우에서처럼 클러스터 크기가 커질수록 대체로 증가하였으나, 8 KB 이하의 구간에서는 그다지 큰 변화를 볼 수 없었고, 그 이상의 구간에서 급격한 증가를 보였다.

2.1.2 FAT 파일시스템에서의 입출력 성능

파일시스템의 성능을 평가하는 데는 여러 가지 관점이 있고, 이 성능을 측정하기 위한 방법 또한 여러 가지가 있으나, 처리량 (Throughput) 중심의 벤치마크 프로그램을 사용하는 방법, 메타데이터 (Metadata-intensive) 중심의 벤치마크 프로그램을 사용하는 방법, 현실 세계의 프로그램을 사용하는 방법 등으로 분류할 수 있다 [6].



(그림 2.4) I/O 처리 시간

이 중 SiSoftware 사의 Sandra 벤치마크 프로그램을 사용하여 현실세계에서 사용되는 디지털 오디오 파일에 대한 파일시스템의 I/O 성능을 측정한 결과는 그림 2.4 에서와 같이 클러스터의 크기가 커질수록 I/O 처리시간은 전반적으로 향상되는 것으로 나타났다.

그러나, 임베디드 오디오 시스템의 경우 임의의 위

치에 대한 쓰기 I/O 는 빈도수가 매우 낮은 동작이므로 성능에 미치는 영향은 작다고 할 수 있다.

또한, 임베디드 오디오 시스템에서 디지털 오디오 파일의 재생에 필요한 연속적인 읽기의 I/O 성능 요구사항은 실질적으로 현재 임베디드 시스템들이 제공하는 최소한의 성능보다 훨씬 낮은 값이므로 성능에 미치는 영향이 작다고 할 수 있다.

따라서, 사용자 입력과 디지털 오디오 특성 값들을 처리하기 위한 임의 위치의 읽기 I/O 에 대한 처리가 중요한 성능요소가 되며, 모바일 디바이스로서 사용되는 경우 사용자로부터의 빈번한 업데이트 등으로 인해 연속적인 쓰기 I/O 에 대한 처리시간이 또한 중요한 성능요소가 된다.

본 논문의 실험에서는 클러스터의 크기가 8 킬로바이트의 구간에서 이 임의 위치의 읽기 I/O 처리 성능이 가장 우수한 것으로 나타났으나, 연속적인 쓰기 I/O 성능은 다른 실험 구간들에 비해 비효율적인 것으로 실험되었다.

실험에 사용된 Test 장비는 삼성전자 데스크탑 PC 제품인 ZCP30 모델을 사용하였으며, 주요 사양은 싱글 펜티엄 4 프로세서, 인텔사의 865GV 칩셋, 512 메가바이트의 ECC PC100 SDRAM, 512 킬로바이트의 캐시메모리, 인텔사의 Extream Graphics2, 삼성 SV4002H IDE 하드디스크, 내장된 10/100 Mbps 이더넷, WindowsXP Pro 서비스 팩 1 등이다. 또한, 사용된 벤치마크 프로그램의 버전은 Sandra Lite Unicode (Win32, x86) 2005.1.10.37 이며, 디스크 캐시와 중첩된 I/O 를 사용하지 않고 하나의 쓰레드 만을 사용하여 연속적인 읽기와 쓰기 및 임의의 위치에 대한 읽기와 쓰기에 대한 I/O 처리시간을 측정하였다.

2.2 플래시 파일시스템과 FTL

플래시 메모리는 일종의 EEPROM (Electrically Erasable and Programmable ROM)으로 크게 바이트 I/O 를 지원하는 NOR 형과 페이지 I/O 만을 지원하는 NAND 형이 있다. NOR 형은 읽기 속도가 빠르나 쓰기 속도가 느려 주로 코드형 메모리로 사용되며 NAND 형은 쓰기 속도가 빠르고 단위 공간당 단가가 낮아 주로 대용량 데이터의 저장장치로 사용된다. 휴대가 용이하고 저전력으로 동작하는 등의 장점으로 인해 플래시 메모리는 디지털 오디오 디바이스의 주 저장매체로 사용되어 왔다.

그러나 데이터를 쓰기 위해서는 비교적 수행시간이 길고 큰 단위로 이루어지는 삭제 연산을 선행해야 하는 등의 설계상 제약이 있으며, 이러한 특성은 플래시 메모리를 주 메모리로 사용하는 것을 어렵게 하고, 보조 기억장치로 사용하는 경우에도 일반 하드디스크용 파일시스템을 그대로 사용하는 것을 힘들게 한다. 또한 기본적으로 NAND 플래시는 양산시 배드 블록 발

생, 최고 10,000 번의 지움 횟수 제한, 기록 및 삭제 에러, 읽기 작업시 1 비트 에러 등 안정성을 저해하는 몇 가지 결함 등이 있다. 따라서 이러한 문제점을 해결하기 위해 파일시스템과 플래시 메모리 사이에 FTL 을 두어 처리하거나 전용의 플래시 파일시스템을 사용하고 있다 [9, 10]

2.2.1 플래시 변환 계층 (Flash Translation Layer)

플래시 메모리의 삭제 연산을 감추기 위한 서비스 층으로 파일시스템과 플래시 메모리 사이에 위치한다. 삭제연산은 쓰기연산 시에 파일시스템이 생성한 논리주소를 플래시 메모리 상의 이미 삭제연산을 수행한 영역에 대한 물리주소로 변환함으로써 감춰진다. 삭제 연산을 감추고 I/O 를 하나의 단위 (Atomic Operation)으로 처리해 하드디스크와 같은 단일 저장장치를 구성함으로써 위층 서비스 층에서 일반 파일시스템을 사용해 플래시 메모리를 제어할 수 있게 한다.

2.2.2 플래시 메모리 전용 파일시스템

JFFS (Journaling Flash File System)과 FFS (Flash File System)과 같은 전용 파일시스템이 GNU GPL 아래 공표되어 주로 리눅스 기반 기기에서 사용되어 왔다.

플래시 메모리 전용 파일시스템과 FTL 을 설계하는데 있어서 성능에 가장 큰 영향을 끼치는 설계 요소는 언제 어떤 블록을 삭제할 것인가 하는 정책결정 과정이며, 너무 일찍 삭제 연산을 수행하면 삭제 블록 내부에 유효한 페이지의 수가 적어 총 수행하는 삭제 연산의 수가 늘어나게 되고, 반대로 삭제연산 수행지점을 최대한 지연시키면 공간 효율성 측면에서는 좋지만 이후에 쓰기 연산이 요청되고 기 삭제된 블록이 없는 경우에 그 동안 지연시킨 삭제연산을 한번에 수행해야 하기 때문에 입출력 성능을 크게 저하시킬 수 있다.

3. 결론

임베디드 오디오 시스템은 파일당 3 ~ 4 분 정도의 재생시간과 128 ~ 192 Kbps 의 샘플링 비율을 가정할 경우 3 ~ 6 메가바이트 크기의 파일들이 대부분을 차지하므로, 이러한 특성에 대한 효율적인 처리가 요구된다. 또한, 디지털 부호화시 수행되는 오디오 샘플링 비율 이상의 입출력 처리 능력이 필요하고, 디지털 오디오 정보들에 대한 빈번한 읽기 동작 등으로 인해 쓰기 보다는 읽기 동작에 대한 성능이 보다 중요해진다.

디스크 타입의 임베디드 오디오 시스템에 대하여 다양한 클러스터 크기에 대한 파일시스템의 처리성과 공간이용률에 관한 본 논문의 시험에서는 공간낭

비율의 경우 클러스터 크기가 커질수록 대체로 증가하였으나, 8 KB 이하의 구간에서는 그다지 큰 변화를 보이지 못하였고, 그 이상의 구간에서 급격한 증가를 보였다. 또한 I/O 처리 성능의 경우 임의 위치의 읽기 I/O 와 연속적인 쓰기 I/O 에 대해서 최선의 값이 다르게 나타났다.

임베디드 시스템에 탑재되는 저장매체로서 2005 년 현재 디스크 타입이 대부분을 차지하고 있으나, 플래시 메모리 타입의 비중이 점점 높아지고 있으므로, 플래시 메모리 타입의 경우에 대해서도 유사한 테스트가 진행되어야 할 것이다.

참고문헌

- [1] In-stat. "Portable Digital Audio Players: Moving to the Beat of a Booming Market". 2005, 06
- [2] Moshe Bar. "Linux File Systems". Osborne
- [3] Seong. "Design and implementation of hard disk drive embeded digital satellite receiver with file management". 2002
- [4] P.J. Shenoy. P. Goyal, S. Rao, and H.M. Vin, "Design and Implementation of Symphony: An Integrated Multimedia File System",
- [5] 이홍재. "Toward understanding Hard Disk". 전자신문사 2003
- [6] Dominic Giampaolo. "Practical File System Design with the Be File System". Morgan Kaufmann Publishers
- [7] Uresh Vahalia. "Unix Internals, the new frontiers". Prentice Hall
- [8] McKusick, M.K. "A fast file system for UNIX". ACM Transactions on Computer Systems 2(3). 1984,08
- [9] Kim, J. M. Kim, S. H. Noh, S. L. Min, and Y. Cho, "A Space-Efficient Flash Translation Layer for CompactFlash Systems," IEEE Transactions on Consumer Electronics, Vol. 48, No. 2, pp.366-375, 2002.
- [10] Samsung Electronics, NAND Flash Memory data sheet, <http://www.samsungelectronics.com/>