대규모 VOD 서버에서의 DISK I/O SUB SYSTEM의 구현

선 창 우*(宣 昌 佑), 최 경 희**(崔 景 熙),정 기 현*(鄭 己 絃)
아주대학교 전기 전자공학부*
아주대학교 정보및 컴퓨터 공학부**
전화 : (0331) 219-2976 / 팩스 : (0331) 212-9531

Implementation of Disk I/O Sub System in Large Scale Video On Demand Server

Changwoo Sun , Kyonghee Choi, Kihyun Chung
School of Electrical and Electronics Engineering, Ajou University
School of Information and Computer Engineering, Ajou University
E-mail : chusun@madang.ajou.ac.kr

Abstract

Video On Demand servers generally require massive disk storages for storing many video data. Many researches have been done on the topics of efficient allocation of movies in disks. This paper, We describe efficient disk placement techniques and implement storage system with SCSI and PCI Bus interface for efficient data handling. This paper also proposes a logical zone reconstruction method for the SCAN data placement technique. The proposed method reconstructs physical zones into logical zones by split and merge operations.

I. 서론

인터넷 상에서 대규모의 주문형 비디오 서버를 구현하기 위해서 필요한 요소는 무엇일까? 우선, 사용자 정보를 감독하고 데이터를 스케쥴링 해 줄 수 있는 management system과, network에 연결할 수 있는 망의 구현, 그리고, 데이터를 셀형식으로 만들어주는 cell generation system의 구현, 그리고 비디오 데이터의 입출력을 위한 DISK I/O system의 구현등 여러 가지가 필요할 것이다. 그 중에서도 본 논문은 이런 종합적인 대규모 VOD 시스템 서버에서 디스크에서의 데이터의 입출력을 위한 시스템을 구현해보고 그에 대한 수행 평가를 알아보고자 한다.

그림 1은 이 연구에 사용된 VOD 서버 시스템의 저장구조 모델을 보여준다. 하나 이상의 SCSI 버스가 각각의 I/O보드에 붙을 수 있고, 하나 이상의 디스크가 각 SCSI버스에 붙을 수 있다. I/O 보드의 버퍼에 있는 내용은 직접 delivery network로 전송되거나, host로 흘러 들어간다. 사용자의 요구와 다른 제어 흐름은 보여지지 않았다.

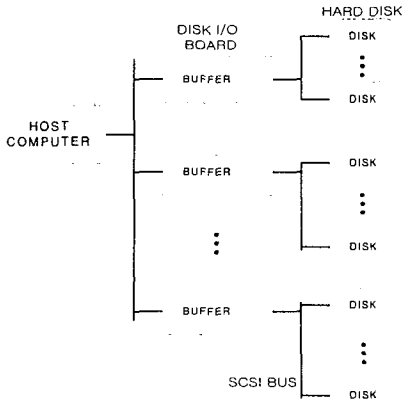

그림 1. VOD 서버의 STORAGE 구조

본 논문에서는 2장에서 PCI BUS와 SCSI BUS와의 연계를 위한 SCSI CONTROLLER를 이용한 I/O SUB SYSTEM의 구현을 보였고, 최대한의 디스크의 출력 대역폭을 위한 존 재구성 방법을 3장에서 보였다. 그리고 4장에서는 486 CPU를 이용한 전체 디스크 I/O system의 구현을 보였다.

II. I/O SUB SYSTEM의 구현

하드디스크의 인터페이스를 위해 본 논문은 SCSI bus를 사용한다. SCSI는 Small Computer System Interface의 약자로 ANSI의 규격으로 정해진, PC와 다양한 주변장치를 잇기 위한 인터페이스 규격이다. SCSI BUS는 여러 가지 면에서 같은 종류의 하드디스크 인터페이스인 IDE 보다 우수하다고 알려져 있다. IDE 버스와 SCSI 버스의 차이점을 표 1에 정리했다.

IDE	SCSI
대용량의 하드디스크를 사용하지 못한다	대용량의 하드디스크를 사용할 수 있다
하나의 컨트롤러에 두 개의 하드디스크만을 달 수 있다	하나의 컨트롤러에 7개에서 15(wide일 경우)개를 달 수 있다.
멀티 스레드가 불가능하다	멀티 스레드가 가능하다
전송속도 최대 5MB/s	전송속도 Ultra Fast 20인 경우 20MB/s

표 1. IDE와 SCSI BUS의 차이점

표 1에서 보여진 SCSI BUS의 여러 우수한 점 때문에 본 시스템에서 SCSI BUS를 적용하였다. 또한, SCSI Controller를 비롯한 여러 주변기기와의 인터페이스를 위해 PCI bus를 사용한다. PCI bus는 CPU에 의존하지 않고 독립된 Chipset으로 버스를 사용할 수 있으며, ISA 등과는 달리 데이터 버스 폭을 32bit 또는 64bit로 사용할 수 있다. 또한, configuration register의 개념을 도입해, plug & play를 가능하게 했으며, 어드레스와 데이터 신호의 멀티 플랙스가 가능하고 버스트로 데이터를 전송할 수 있다. PCI bus의 밴드 폭은 32bit의 데이터 버스에서 132MByte/sec의 속도로 빠른 전송을 할 수 있다. 이상 위에서 보여진 이점 때문에 본 논문에서 보여진 I/O sub system에는 PCI bus를 사용한다. 주변기기와의 연계를 위한 PCI bus와 하드디스크와의 연계를 위한 SCSI bus의 사용을 만족하기 위해 Qlogic에서 나온 ISP1040C SCSI controller processor를 사용하였다. ISP 1040C를 이용하여 구현한 I/O sub system의 구조는 다음 그림과 같다.

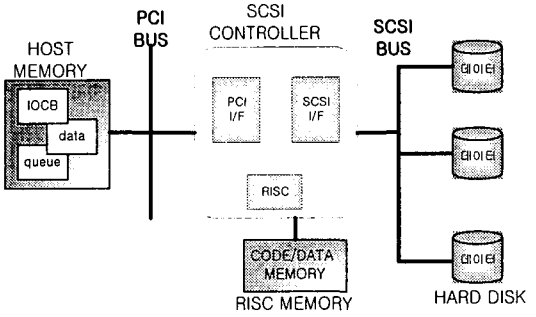

그림 2. I/O Sub System의 구현

SCSI controller는 자체 메모리 영역을 갖고 있어서 내부 RISC processor의 firmware code를 저장한다. SCSI controller에 있는 PCI interface로 PCI bus에 직접 연결할 수 있고, 마찬가지로 SCSI interface부분으로 하드디스크와 직접 연결이 가능하다. SCSI controller는 호스트 프로세서를 간섭하지 않고 오퍼레이션을 수행할 수 잇다.

III. Disk Reconstruction Algorithm

하드디스크의 속도를 결정하는 disk access time은 탐색시간, rotational latency, 미디어 전송시간, 헤드 스위칭과 실린더 스위칭 시간과, 버스 할당, 버스 전송시간이 중요한 요소가 된다. 그 중 헤더 탐색 시간이 가장 긴 시간을 갖는다. 이러한 헤더 탐색 시간을 짧게 하기 위해서 본 논문에서는 Circular SCAN알고리즘의 disk scheduling model을 사용한다. 현대의 대용량 하드디스크는 디스크의 영역을 존으로 나누고, 상대적으로 데이터의 전송속도가 빠른 바깥쪽 존에 섹터의 수를 많이 할당하여, 효율적인 데이터 엑세스를 가능하게 하였다. 이러한 구역 구조 디스크 특성을 최대한 살리고, 최적의 서비스를 위한 방법으로 본 논문에서는 디스크 bandwidth를 개선하고, 구역 구조 디스크에서의 낭비되는 디스크 space를 줄이기 위해 디스크의 병합(merge)과 분할(split)과정을 통해 논리적으로 재구성하는 방법을 제안한다.

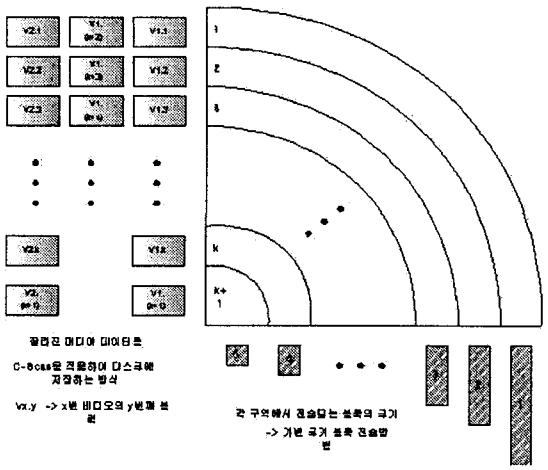

그림 3. 디스크 존에서의 비디오 데이터 배치방법

그림 3은 실제로 하드디스크 안에서 비디오 데이터가 어떻게 배치되는 지를 보이고 있다. 그림의 부채꼴 모양이 존으로 나뉘어진 하드디스크의 단면이고, 왼쪽의 셀 부분이 잘려진 비디오 데이터들이다, 그리고 밑 부분의 셀은 존별로 잘려진 비디오 데이터의 셀의 크기를 나타내고 있다. 비디오 데이터는 그 용량이 크기 때문에 Working Set이라는 하나의 단위로 잘려지고, CSCAN 알고리즘에 의해 바깥쪽 존에서 안쪽 존으로 순서대로 채워진다. 존마다의 전송속도가 다르기 때문에, 각 존마다의 데이터를 읽는 시간을 맞추기 위해 가변 크기 블록 전송방법을 사용하였다.

가변 크기 블록 정책을 이용한 SCAN 알고리즘의 요구에 부합하기 위해서, 모든 존의 전송률과 저장용량의 비율은 고정적이어야 한다. 즉, 이상적인 존에서의 가변 크기는 다음 (식1)을 만족해야 한다. 이상적인 i번째 블록을 Z'_i 라고 하자. 각 블록의 전송시간이 가변 크기 블록 사이즈에서 모든 존에 동일하도록 요구된다. 즉,

$$\frac{B(Z'_1)}{R(Z'_1)} = \frac{B(Z'_2)}{R(Z'_2)} = \cdots = \frac{B(Z'_K)}{R(Z'_K)} = T_{disk}(Z_j), \quad \forall j \in (1,k)$$

(식.1)

여기서 $B(Z'_i)$는 재구성된 i번째 존에서의 블록 사이즈를 말하고, $R(Z'_i)$는 재구성된 i번째 존에서의 전송속도를 의미한다. T_{disk}는 각 블록을 읽는 데 걸리는 시간을 의미한다. 각 존에 할당된 데이터의 수(N)는 SCAN 알고리즘에서 고정적이고, 이상적인 존에서의 어떠한 디스크의 낭비도 없다면, 다음 (식2)를 만족하게 된다.

$$\frac{S(Z'_1)}{R(Z'_1)} = \frac{S(Z'_2)}{R(Z'_2)} = \cdots = \frac{S(Z'_K)}{R(Z'_K)} = C \qquad (식.2)$$

C는 이상적인 경우의 존의 크기에 대한, 전송률의 일정한 비율을 나타낸다.($C = T_{disk} * N$), $S(Z'_i)$는 이상적인 구역의 크기이다. ($S(Z'_i) = N * B(Z'_i)$)

통상적인 디스크의 물리적 존은 위의 수식과 확실히 일치되지 않기 때문에, 요구를 만족시키기 위해서 본 논문은 물리적인 존을 적절하게 분할, 또는 병합시키는 방법으로 논리적 존으로 수정한다. (식2)에 의해,

$$S(Z'_i) = R(Z'_i) \times C \quad (식\ 3)$$

$\sum_{i=1}^{K} S(Z_i') = C\sum_{i=1}^{K} R(Z_i') \le \sum_{i=1}^{K} S(Z_i) = T_{ave\_svr}\sum_{i=1}^{K} R(Z_i)$ 이므로,

$$C = \frac{\sum_{i=1}^{K} S(Z'_i)}{\sum_{i=1}^{K} R(Z'_i)} \le T_{ave\_svr} \qquad (식.4)$$

$R(Z_i)$는 물리적 영역에서의 존 i에서의 전송률이고, $S(Z_i)$는 물리적 영역에서의 존 i의 크기이다. T_{ave_svr}는 각 물리적 구역에서의 평균 서비스 시간을 의미한다. (식4)에서 C의 최대 값은 전체 디스크의 평균 서비스 시간이 된다.

본 논문에서는 타겟 디스크 모델로 HP C2247을 사용한다. 존 재구성을 하기 위해 물리적 존의 첫 번째 구역부터 재구성을 시작한다. 이미 알려진 데이터로, C의 값을 구할 수 있고, C와 첫 번째 구역의 전송률을 통해 (식3)에서 보여진 대로 첫 번째 존의 이상적 구역의 크기인 $S(Z'_1)$을 구할 수 있다. 이상적인 구역이 만들어진 후에 첫 번째 물리적 존이 아직 남았다면, 같은 방법으로 $S(Z'_2)$를 만들 수 있고, 영역이 모자란다면, 모자란 영역 전체를 다음 존과 합하여, 다음 존의 전송률을 적용하여, 재구성을 한다.

그림 4는 HP C2247의 물리적 존과 알고리즘을 적용하여 재구성된 존을 보였다.

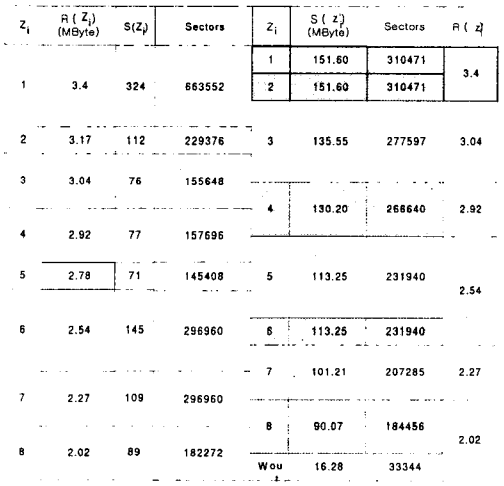

그림 4. HP C2247의 물리적 존과 재구성된 존

그림에서 W_{out}은 구역 재구성으로 인해 낭비된 디스크의 공간이다. 물리적 영역에서의 비디오 데이터를 CSCAN 방식으로 저장했을 경우에 생기는 낭비된 영역이, 구역 재구성으로 인해 현저하게 줄어들었다. 비디오 서버가 일반적으로 많은 양의 프로그램들을 포함하고, 많은 수의 디스크를 갖고 있음을 고려해 볼 때, 사용자 수의 증가와 각 디스크에서 낭비되는 영역의 절약은 현저한 performance의 증가를 가져올 것이다.

Ⅳ. 시스템의 구현

본 논문은 486 base system에서의 디스크 I/O system을 제안한다. 실제로 본 논문에서 보여진 DISK I/O system이 사용될 대규모의 비디오 서비스에서 management system은 pentium 급의 고속 micro processor를 사용했지만, 상대적으로 디스크 I/O에서는 486으로도 충분히 입출력을 담당할 수 있기 때문이다. 차후 다시 고속의 data 처리가 필요한 경우 486 cpu에서 pentium 급으로 업그레이드 할 수도 있다.

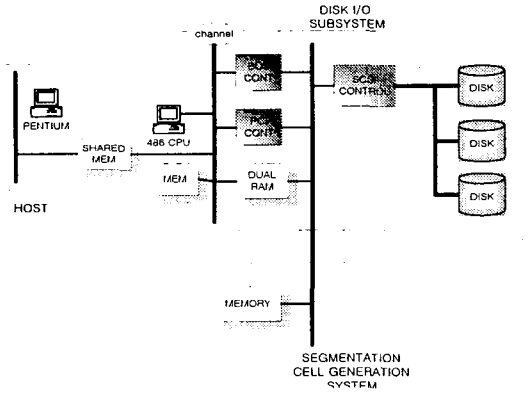

그림 5 DISK I/O SYSTEM의 구현

그림 5는 최종적으로 구현된 DISK I/O system을 보였다. Channel 부분을 만들어, 보드의 프로그램을 모니터 할 수 있게 하였고, system bus를 control하기 위한 bus controller, PCI bus를 control 하기 위한 PCI bus controller, SCSI bus를 control 하기 위한 SCSI bus controller (ISP1040C) 을 달았다. 또한, CPU와 DISK와의 데이터를 주고받기 위해, 다리 역할을 해줄 dual ram을 만들어 CPU의 local 버스에서 access하고 PCI bus에서도 access할 수 있게 만들었다. 또한, DISK의 데이터를 PCI 버스에 실어 cell generation, 또는 특정한 역할을 할 어떤 다른 보드에 보내기 위해 memory를 만들어 전송할 수 있게 하였다. DISK I/O system과 host(management system)와의 통신을 위해 또한, shared memory도 구현하였다.

Ⅴ. 결 론

본 논문은, 제안된 방법으로 디스크를 물리적 구역에서 논리적 구역으로 재구성하여, 비디오 서버에서의 디스크의 낭비를 최소화 할 수 있다는 것을 보였다. 또한, 실제로 SCSI와 PCI bus를 이용한 디스크 I/O system도 구현하였다. 논문에서 제안한 방법으로 시스템을 구현하여, 실제로 디스크의 입출력을 할 수 있는 단계까지 실험이 전개되었고, 추후, 실험을 통해 디스크을 논리적으로 재구성해 수행 결과 만족할 만한 결과값이 나오리라 예상된다 일반적으로 Video On Demand 서버가 매우 많은 수의 비디오 프로그램을 포함한다는 것을 고려해 볼 때, 제안된 방법이 대용량의 저장매체를 갖고 있는 구역 구조 디스크를 보유한 VOD서버에서 이용하는 방법에는 더 효율적인 디스크의 이용 방법임을 보이고 있다

참고문헌(또는 Reference)

[1] C.Wu,et.al., " A Scalable architecture for video-on-demand server", IEEE Trans. on Consumer electronics, Vol.42, NO.4 pp.1029-1036 Nov. 1996

[2] Antoine N.Mourad, "Issues in the de/sign of a storage server for video-on-demand", Multimedia Systems, No.4 pp.70-86 1996

[3] Qlogic, "Intelligent SCSI Processor", Qlogic Corporation, December 1993

[4] Tom Shanley and Don Anderson, "PCI System Architecture",Mindshare,1993

[5] Shahram Ghandeharizadeh, "Design and implementation of scalable continuous media servers",Parallel Computing, Vol.24 1998 pp.91-122