2001년 한국정보처리학회 춘계 학술발표논문집 제8권 제1호

분산 공유 디스크를 위한 고 가용성 디스크 캐쉬 관리자에 관한 연구

o
진광윤**, 한 판암*
**삼척대학교 컴퓨터공학과
*경남대학교 컴퓨터공학과
e-mail:kyjin@samchok.ac.kr

A study on High-availability Disk Cache Manager for distributed shared-disk

Kwang-Youn Jin**, Pan-Am Han*
**Dept of Computer Engineering, Samchok University
*Dept of Computer Engineering, Kyoungnam University

요 약

본 연구는 다중 PC 노드로 연결된 클러스터링 시스템을 위해 마이크로 커널을 기반으로 한 분산 운영체제를 탑재하여 운영되는 DCM의 고 가용성과 공유 디스크의 입출력 수행 속도를 향상시키는 데 있다. 마이크로 커널에서 제공된 서로 다른 디스크를 메시지 패싱 기법을 이용하여 입출력 성능 향상과 디스크 상의 자료 무결성을 보장할 목적으로 고 가용성의 DCM를 설계 및 구현하여 시스템의 생산성을 향상시킨다.

1. 서론

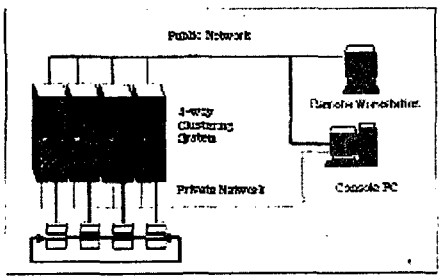
클러스터링(Clustering) 시스템은 다양한 응용을 요구하는 대단히 큰 문제를 여러 개의 수행 단위를 각 노드(PN)인 클러스터등에 분해하여 동시에 수행케 함으로써 빠른 결과를 얻을 수 있도록 설계된 약결합(Loosly-Coupled) 병렬 처리 구조이다. 그러나 데이터 의존형 응용 분야의 작업에서는 동시성과 병렬성에 장애가 되기 때문에 이에 대해 특별한 고려가 요구되기도 한다[1]. 클러스터링 시스템은 병렬처리를 위한 기술로 비교적 값이 싼 컴퓨터들을 네트웍으로 연결하여 전체가 하나의 고성능 서버처럼 동작하게 하는 기술로서 시스템 구축비용이 적게들고, 보다 큰 시스템으로 확장이 용이하며 가용성이 높다는 장점을 가지고 있다. 이러한 이점 때문에 높은 연산 능력을 요구하는 문제를 해결하고, 또한 인터넷 대용량 웹 서버를 지원하기 위해 사용되어지고 있다. 따라서 본 연구에서는 차세대 인터넷 서버의 한 방안으로 클러스터링 시스템이 채용되었을 때, 입출력 성능을 향상하고 디스크 자료의 무결성을 보장할 목적으로 마이크로 커널 기반 메시지 전달방식으로 동작하는 디스크 캐쉬 관리(DCM; Disk cache Manager)를 제안하고자 한다. 제안하는 DCM 은 디스크 캐쉬의 효과뿐만 아니라 공유 디스크 관리 문제에 많은 비중을 두고 있으며, 공유 디스크를 두개의 논리 디스크로 나누어 캐싱하므로써 디스크 데이터 공유로 인하여 발생되는 동시 접근은 자료의 일관성 문제점을 보완하여 노드가 통신부하를 가하지 않고 공유 디스크를 실현할 수 있으며, 병목 현상 등으로 느린 입출력 성능을 극복하는 방법으로 운영체제의 파일 관리자(File Manager)로부터 입출력 시 두 논리 디스크로 두개의 데이터 경로를 두어 병렬 입출력 처리가 가능하도록 제안한다.

2. 시스템 구성

2.1 시스템 하드웨어

본 연구에서 제안하는 DCM의 수행 환경은 공유와 비 공유 디스크 구조에 대한 어느 특정 한 하드웨어에 종속성을 가지지 않으며, 공유와 비 공유 디스크를 지원하기 위하여 2가지 논리 디스크를 정의하고 각각 캐싱이 가능하게 운영한다. DCM이 운영되는 시스템 하드웨어의 구성은 [그림 2-1]에서 제

시된 것과 같이 4대의 PC를 Ethenet으로 결합하여 하나의 클러스터링 시스템으로 구성된다. 각 PC는 클러스터에서 하나의 노드로 정의하여 공용 네트웍 상에서 공유되는 DB 서버로서 주로 입출력 병목현상의 해결을 원하는 입출력 지향적인 작업을 수행하는 시스템으로 운용되어 진다. 시스템에서 각 노드는 시스템 메모리의 일정 용량을 제외하고는 디스크 캐쉬로 사용하며, 시스템의 고 가용성 향상에 대해서는 시스템 내에서 단일 장애점을 없애기 위해서 Battery Back-up 장치를 내장하여 고장 혹은 전원 차단에 대하여 기존의 자료를 보존할 수 있도록 구성한다. 또한 각 디스크의 자료 보존과 어떠한 장애의 가능성에 대한 대비와 언제라도 완벽한 유지보수와 결함허용과 고 가용성의 이중화를 위해 이중 포트(Dual Port) 디스크를 지원하고, 각 디스크는 광 채널 혹은 SCSI에 의해 각 노드에 의하여 공유 되도록 접속된다. 하드웨어 구조는 RAID 1, 5를 지원하는 disk array 형태인 20개까지의 디스크 스트링으로 구성이 가능하다[5].

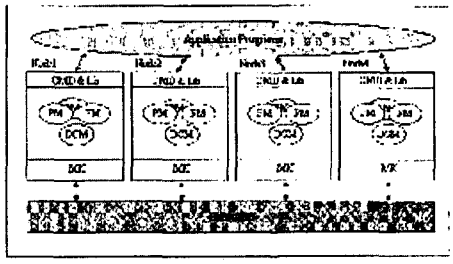


[그림 2-1] 목표 시스템 구조(4-Way PC 클러스터링 시스템)

2.2 시스템 운영체제

본 연구에서 가정하는 운영체제는 마이크로 커널 기반 위에 유닉스 운영체제를 기능적으로 서버화 한 Chorus 마이크로 커널 기반의 병렬 운영체제인 MISIX 구조이다[5]. 각 노드에는 Chorus 마이크로 커널 상에 유닉스의 기능적 서버인 프로세스 관리자(PM), 파일 관리자(FM), 블록 입출력 관리(BIOM) 등을 가지며, 본 연구에서 제안하는 DCM을 추가하여 전체 입출력의 기능을 향상시킬 수 있는 서버로서의 구성을 갖도록 하였다. 각 노드에 연결된 유닉스 서버는 마이크로 커널의 지원을 받아 Chorus IPC 메커니즘인 포트 통신에 의해 메시지 전송을 수행하며 동일 서버들 간의 일관성 유지와 수행 동기화를 추구한다. 물리적으로 여러 개의 운영체제를 가지지만 논리적으로 Single System Image(SSI)를 제공하며 사용자에게는 시스템 전역적인 병렬처리 환경을 지원한다. [그림 2-2]는 MISIX 주요 서버들의 노드간의 배치 및 관계를 도시하고 있다[2][3][6]. FM은 운영체제 버퍼 캐쉬에서 실패(miss)하는 모든 데이터 블록에 대해 일치하는 캐쉬 라인으로 변환하여 DCM에 메시지로 요청한다.

DCM은 디스크 캐쉬에서 해당 블록이 성공(hit)하면 FM의 요청을 처리하고 실패(miss)하면 장치 구동기를 관리하는 Block I/O Manager(BIOM)에게 물리적인 입출력을 동반한 처리를 요청한다.



[그림 2-2] 운영체제의 노드간 서버 구성도

2.3 고 가용성 입출력 구조

본 연구에서 입출력 자료의 무결성을 보장하기 위하여 결함허용 파일 관리자(FTFM)를 두고 있다[5]. FTFM은 제안하는 DCM의 효과적인 운용을 위하여 각 노드의 기능적 서버로서 동작하며 본 연구에서 제안하는 범위의 밖에 있다. 따라서 제안하는 DCM의 고 가용성 관점에서 FTFM을 기술하고자 한다. 결함허용 파일 시스템은 파일을 관리하는 프로세스를 이중화하는 방식을 사용한 것이다. 결함허용을 위하여 동일한 서비스를 수행하는 FTFM를 서로 다른 노드에 두며, 하나의 프로세스는 실제로 서비스를 제공해주는 primary로 다른 하나의 프로세스는 primary가 오류를 발생할 때까지 기다리는 backup으로 동작하게 되는 Host-standby 방식을 채택하였다. 이를 실제로 구현하기 위해서는 FTFM의 primary와 backup간의 동기화 문제를 해결해야 하며, 그리고 서비스를 받는 클라이언트들이 primary의 서비스가 중단되더라도, 서비스를 요청하는 포트(port)는 기존과 변함이 없이 사용할 수 있어야 한다. 그리고, primary나 backup에 오류가 발생하였을 때에, 이를 상대방에게 알려줄 수 있는 방법이 있어

야 하는데, 이를 해결하기 위하여 안정영역(Stable Area),포트가명(portAlias), 그리고 결함 관리자(Fault Tolerant Manager)에 대해서도 앞으로의 연구 과제가 되겠다[5].

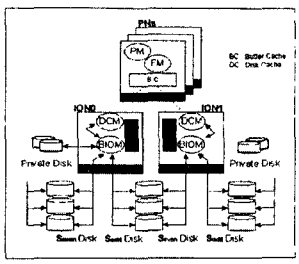
3. DCM 설계

3.1 DCM 구조

DCM은 Chorus MK상에서 메시지 통신으로 다른 서버와 통신하는 시스템 서버(Actor)로서 자신의 번지 공간과 자원을 소유한다. DCM은 다른 서버와의 통신을 위해서 포트(Port)를 가지며 각 Port에 접속된 메시지를 수신하여 처리하는 부분과 디스크 캐쉬 관리 부분으로 나뉘어 진다.

3.2 캐쉬 관리 모형

운영체제 파일 관리자(FM)의 버퍼 캐쉬 실패로 인해 물리적 입출력이 요구되면 IPC를 통해 인터페이스 된다. ION(Input Output Node)에 장착된 지역 디스크를 *Private Disk*로 정의하고, 공유 디스크를 두 개의 논리 디스크인 S_{even} Disk와 S_{odd} Disk로 정의하여 캐싱 영역을 관리한다. [그림 3-1]은 2단계 캐쉬 관리 모형으로서 두 ION이 공유하는 디스크와 인접 ION이 공유하는 디스크 및 *private disk*의 관계와 PN상에 존재하는 FM의 버퍼 캐쉬(1차 캐쉬) 및 디스크 캐쉬(2차 캐쉬) 등의 관계를 표현하고 있다. 공유 디스크에 대해 FM의 1차 캐쉬 실패는 FM의 분배기(를 거쳐 해당 노드의 2차 캐쉬 관리자인 DCM에 병렬 입출력 요구 메시지를 보낸다.



[그림 3-1] DCM 적용 시스템 모델

3.3 캐쉬 관리 정책

기본적인 캐쉬 관리 정책으로 WB(Write-Back)을 제공한다. WB는 읽기 시에는 디스크 캐쉬로부터 읽게 하고, 쓰기 시에는 디스크에 즉시 쓰지 않고 디스크 캐쉬에 쓰기를 한다. 또 다른 캐쉬 관리 정책으로 WT(Write-Through)을 지원한다. WT는 읽기 시에는 디스크 캐쉬를 참조하나 쓰기 시는 즉시 디스크에 쓰기를 하는 방식이다. WT의 쓰기 또는 캐쉬 실패(miss)때는 물리적 입출력 요청을 위하여 장치 구동기 서버인 BIOM으로 메시지를 전달한다.

3.4 캐쉬 관리 구조

캐쉬 관리구조는 캐쉬를 관리하기 위하여 해쉬 테이블(Hash table)을 사용하며 캐쉬 세그먼트를 해쉬하는데, 캐쉬 접근을 위하여 디스크 번호와 블록 번호를 태그(tag)로 사용한다.

(1) Cache segment

캐쉬 세그먼트는 캐쉬 관리를 위한 단위로서 정보 파트와 데이터 파트로 구성되었다. 장치번호와 블록 번호로 구성된 Tag, 캐쉬 해쉬표를 연결되는 전, 후방 포인터, 사용가능영역인 *freelist*에 연결되는 전,후방 포인터, 캐쉬 라인 및 블록의 상태 휠드 그리고 디스크 캐쉬영역인 라인을 가리키는 포인터 등으로 구성된다.

(2) Cache line

캐쉬 라인은 디스크로부터 데이터 블록을 패치(fetch)할 때 가장 효과적인 단위로서 연속적인 8개의 섹터로 구성하였다.

(3) Cache Block

캐쉬 블록은 일관성 유지를 위한 단위이고 운영체제가 요구하는 최소 블록 규격이다, 8개의 캐쉬 블록은 한 라인을 형성하고, 실제 디스크 상에 물리적으로 연속적이다.

(4) Hash Table

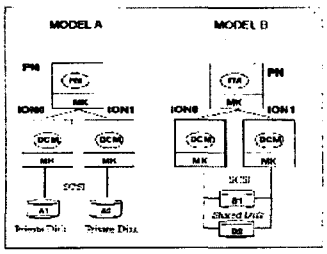
캐쉬 구조를 형성하는 테이블은 캐쉬 세그먼트 태그 휠드에서 장치 번호와 블록번호의 각각에 대하여 테이블을 가진다. 장치 번호는 두 ION이 공유하는 장치를 구별하기 위하여 필요하며 ION 관점에서 볼 때 다음과 같은 세가지 논리적인 장치를 정의한다.

- *private disk* : 해당 ION에 연결된 local disk
- S_{even} *disk* : 공유 디스크의 EVEN 라인 영역
- S_{odd} *disk* : 공유 디스크의 ODD 라인 영역

4. 구현 환경 및 결과

4.1 구현 환경

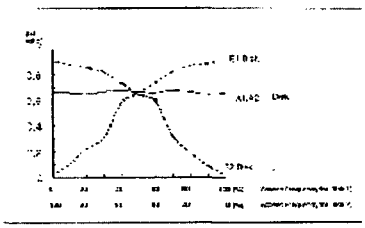
DCM을 테스트하기 위하여 3개의 펜티움 PC로 테스트 베드를 구성하였다. 한 대의 PC에는 어플리케이션 서버로서 처리 노드(PN) 역할을 수행하도록 FM을 포함한 MISIX의 기능적 서버들이 탑재되었고, 2대의 PC는 입출력 노드(ION) 역할을 하기 위하여 DCM과 BIOM을 두었다. 각 PC에는 Chorus MK 기반의 유닉스 기능적 서버인 FM, FTFM, CONM 등이 공통적으로 탑재된다. 각 노드들을 이더넷에 접속하여 3-way 클러스터링 시스템을 형성하고 있다. 본 실험에서는 DCM의 효과를 측정하기 위하여 각 노드의 개별 디스크(private disk)로 운영하는 MODEL A와 각 노드가 디스크를 공유하는 MODEL B로 나누어 [그림 4-1]과 같이 시스템을 구성하였다.



[그림 4-1] DCM의 구현 모델 구조

4.2 구현 결과

각 모델의 디스크 1,2에 일정한 입출력 부하를 걸어 그 변화에 대한 적중률(Hit Ratio)을 측정하였다. 이때 두 디스크의 걸리는 입출력 부하의 합은 전체 부하에 대한 100%를 넘지 않게 한다.



[그림 4-2] 디스크 사용량에 따른 캐쉬 적중률

DCM 커널에 캐시 적중(cache hit)과 캐시 실패(cache miss)를 위한 전역 카운터 변수를 삽입하여 각 백분률에 대한 적중률을 저장하여 구한다. 두 모델에 동일한 시험 프로그램을 수행하여 [그림 4-2]와 같은 결과는 얻었다. A1과 A2 디스크는 비공유 방식으로 상호 독립적이기 때문에 부하의 정도에 관계없이 거의 같은 적중률을 보였다.

5. 결 론

본 연구에서는 마이크로 커널 상에 수행하는 병렬 운영체제에서 하나의 기능적 서버로서 디스크 캐쉬 관리자(DCM)를 설계 및 구현한 내용을 기술하였다. DCM 설계의 결과로서 Chorus IPC 규격에 맞게 서버화 하였고, 캐쉬의 효과를 극대화하기 위해 먼저 최적화된 물리적 입출력 단위를 캐쉬 라인으로 정의하였으며 한 라인을 운영체제가 요구하는 최소의 단위인 블록으로 하는 2단계 캐쉬 관리 기법을 제안하였다. 또한 ION에 장착된 장치를 3개의 논리 장치로 구분, FM의 분배기를 통하여 *Even/Odd* 캐쉬 라인에 대한 병렬 입출력이 가능하고, ION 고장에 대한 교체 경로를 제공함으로써 입출력 성능 향상과 고장 시에 유연하게 대처하게 하였다. 향후 본 연구는 두 개의 ION에 접속된 공유 디스크 문제를 확장하여 n개의 ION과 m개의 디스크를 함께 접속하여 *mod(n)*개의 논리 디스크에 대한 연구를 계속 수행할 것이다.

참고 문헌

1. H.J.Kim, K.W.Rim,"The Design and Implemen-tation of a Parallel Processing Operating Syatem for Massively Parallel Processing Computer",Proceeding of ITC-CSCC,pp.645-648, july 1996.
2. Y.J.Nam, J.M.Kim,"The Design and Implementa-tion of the Terminal Device Driver Server on top of Microkernel for SPAX",Lecture Note in Computer Seience:HPCN,vol 1067, pp.1009-1010, April 1996.
3. J.M.Kim, Y.J.Nam, H.J.Kim,"The Design and Implementation of a Disk Cache Manager for a MK-based Parallel OS,"Proceeding of the IASTED SE'97,pp88-92,Nov 1997.
4. M.Rozier, et al.,"Overview of the Chorus Distributed Operating System,"Proceeding of USENIX Workshop on Micro Kernel and Other Kernel Architectures,pp.39-69,April 1992
5. Abraham Siberschatz, Peter Baer Galvin,"Opera-ting System concepts",4th Edition, John Wiley & Sons Ltd, pp568-589, 1998.
6. 김 주만,"고속 병렬 컴퓨터를 위한 고장 감내 파일 시스템 설계", NCS 98, Vol 1 pp216-219,1997.