제 18 회 한국정보처리학회 추계학술발표대회 논문집 제 9 권 제 2 호 (2002.11)

iBASE/Cluster: 클러스터 환경을 위한 바다-IV 의 확장[1]

김홍연, 진기성, 김준, 김명준
한국전자통신연구원
e-mail : {kimhy, ksjin, jkim, joonkim}@etri.re.kr,

iBASE/Cluster: Extending the BADA-IV for a Cluster environment

Hong-Yeon Kim, Ki-Sung Jin, June Kim, Myoung-Joon Kim
Electronics and Telecommunications Research Institute

요 약

iBASE/Cluster 는 한국전자통신연구원에서 개발한 객체 지향 멀티미디어 DBMS 인 바다-IV 를 기반으로 공유 디스크 기반의 클러스터 환경에서 무정지 DBMS 서비스를 지원하기 위하여 확장된 DBMS 이다. 본 논문에서는 이를 위하여 수행된 바다-IV 의 확장에 대해 기술한다. 기술된 내용은 공유 디스크 자원 접근을 동기화 하기 위한 동시성 제어 기법 및 버퍼 관리 기법, 그리고 특정 노드에 고장 발생 시 회복 기법 등에 대한 확장을 포함한다. 또한 확장된 DBMS 가 LVS(Linux Virtual Server) 기반의 클러스터에서 자동적인 고장의 감지 및 이전(fail-over)을 수행 하기 위한 연동 구조를 다루고 동작 시나리오를 통하여 구현된 시스템의 흐름을 보인다.

1. 서론

최근 인터넷 환경에서 급속히 증대되는 24 시간 무정지 서비스 요구를 효과적으로 처리하기 위하여 고가용을 위한 클러스터 컴퓨팅 시스템의 활용이 대안으로 시도되고 있다. 이러한 시도는 클러스터 노드, 연결망 등의 하드웨어 수준[1,2], LVS[3], Sun Cluster Solution[4] 등의 운영 체제 수준, GFS[5], Sanique[6], SANtopia[7] 와 같은 파일 시스템 수준, Oracle 9i[8], IBM DB2[9]와 같은 DBMS 수준 등의 다양한 수준에서 이루어 지고 있다.

특히 Oracle 9i, DB2 등은 비공유(shared nothing) 환경뿐 아니라 공유 디스크(shared disk) 환경에서 운영을 목적으로 개발 되었으며 전역 잠금 관리, 전역 버퍼 관리, 회복 등 클러스터 DBMS 를 위한 연구[10,11,12,13]에 많은 진척을 보이고 있다.

한국전자통신연구원에서도 이러한 시도의 일환으로 단일 시스템 환경을 위해 개발 되어진 바다-IV DBMS[14]를 고가용 구조의 클러스터 환경에서 운영 될 수 있도록 확장하기 위한 연구가 수행 되었다.

본 논문은 이 과정에서 이루어진 분석, 설계 , 구현 등의 주된 이슈를 다룬다. 본 논문에서 다루는 주된 내용들은 프로세스 구조, 전역 동시성 제어, 버퍼 일관성 제어, 전역 회복, 고장 이전(fail-over) 지원 기능 등을 포함한다.

본 논문의 구성은 다음과 같다. 먼저 2 장에서 본 논문의 가정 및 요건에 대해 살펴보고 3 장과 4 장에서 바다-IV 를 클러스터 환경을 위해 확장한 내용과 이를 기반으로 구현된 iBASE/Cluster 를 기술한다. 마지막으로 5 장에서 결론 및 향후 연구 방향을 다룬다.

2. 가정 및 요건

본 장에서는 클러스터 환경에서 운영을 위한 바다-IV DBMS 의 확장에 있어서 본 논문에서 가정한 사항들과 확장된 바다-IV DBMS 인 iBASE/Cluster 가 제공해야 하는 기능에 대해 기술한다.

- o 바다-IV DBMS 를 클러스터 환경으로 확장한다. 따라서 엄정 2 단계 잠금, 로그 기반 회복 기술을 계속 사용한다.
- o 확장된 클러스터 DBMS 는 공유 디스크(shared disk) 클러스터에서 운영한다.
- o 확장된 클러스터 DBMS 는 클라이언트에게 단일 DBMS 로 간주 될 수 있도록 단일 시스템 이미지(single system image)를 지원해야 한다.
- o 확장된 클러스터 DBMS 는 바다-IV 클라이언트를 수정 없이 사용할 수 있어야 한다.
- o 확장된 클러스터 DBMS 는 Active/Active 방식의 고장 이전 및 회복을 지원한다. 모든 노드가 정상일 경우 클라이언트는 모든 노드에 공평하게 분배된다. 특정 노드에 고장

[1] 본 논문은 정보통신부 지원으로 2001 년부터 2 년간 수행하는 "클러스터 기반 DBMS 기술 개발" 과제의 연구 결과임.

이 발생한 경우 정상인 노드들이 클라이언트들을 분배하여 처리하며 고장 발생한 노드에서 수행 중이던 트랜잭션은 모두 로그를 기반으로 철회한다.

o 현재의 확장은 자료저장시스템 수준의 확장으로 한정한다. 따라서 클러스터는 자료저장시스템 수준에서 병렬성을 확보하기 위해 사용되며 병렬 질의 수행(intra-query *parallelism*)은 지원하지 않는다.

3. 바다-IV 확장

3.1. 프로세스 구조 확장

기존 구조는 클라이언트와 바다-IV DBMS 간에 1:1 프로세스 구조이다. 이를 위하여 한 개의 바다-IV 데몬 프로세스가 상주하여 클라이언트의 접속 시 대응되는 서버 프로세스를 기동(fork) 시키는 구조이다. 기동 된 서버 프로세스는 서버 데몬의 개입 없이 클라이언트와 직접 소켓을 통해 교신하며 운영된다.

클러스터 환경에서 운영을 위해서는 기존 구조가 모든 노드에서 서비스 가능하도록 확장 되어야 한다. 이를 위해 모든 노드에 상기한 바다-IV 데몬 프로세스를 상주 시켜 클라이언트가 어떤 노드로 접속하여도 서비스를 받을 수 있도록 확장 한다.

3.2. 단일 이미지 지원

다수의 노드에서 수행되는 바다-IV 를 하나의 단일 DBMS 로 간주되도록 하는 단일 이미지를 지원하기 위해 클라이언트의 바다-IV 접근 방법을 확장한다. 이를 위하여 가상 IP 에 기반 하는 Linux Virtual Server(LVS)를 사용한다. 클러스터는 가상 IP 를 가지며 모든 클라이언트가 이 가상 IP 로 접근하면 이를 실제 노드로 중계한다. 노드에 고장이 발생하면 그 노드는 LVS 수준에서 분배 대상에서 제외 된다. 이 과정은 클라이언트 소프트웨어에 투명하게 일어나며 언제나 가상 IP 로 접근하기만 하면 충분 하므로 기존의 클라이언트 API 는 확장할 필요가 없다.

3.3. 공유 자료 접근 방법 확장

바다-IV 에서 트랜잭션들은 파일, 색인, 카탈로그 등의 공유 자료에 접근하며 수행한다. 이들에 대한 동시성 제어를 위해 잠금, 래치, 버퍼 등을 사용한다. 이러한 방법들은 클러스터상에서 적용하기 위해 확장 되어야 한다.

먼저 기존 잠금 관리를 클러스터 노드간 잠금을 지원하기 위해 전역 잠금 관리로 확장한다. 각 노드의 모든 잠금 요청은 네트워크 등의 경로를 통해 전역 잠금 관리기로 전달 되어 처리된다.

또한 모든 잠금 요청이 원격지로 전송 됨으로 인한 성능 저하를 방지하기 위해 잠금 권한 캐쉬(lock authority caching) 기법을 사용한다. 이는 한번 획득한 잠금은 트랜잭션이 반납 하더라도 그 노드에 계속 캐쉬 시키는 방법이다. 잠금 권한이 캐쉬 되어 있으면 그 잠금에 대한 처리는 전역 잠금 관리자의 개입 없이 그 노드에서 처리할 수 있다. 이를 위해 지역 잠금 관리기를 사용한다. 이러한 구조는 노드 내 잠금의 지역성(locality)에 기반 하여 원격 잠금 요청 회수를 줄이는 효과를 가진다.

버퍼는 노드간 버퍼 일관성 제어 문제를 해결하도록 확장 한다. 단 접근 하려는 자료의 종류 및 잠금 단위에 따라 적용 가능한 일관성 유지 방법이 다르다.

카탈로그 페이지 및 색인 페이지에 대해서는 버퍼 관리기 수준에서 회피(avoidance) 기반 일관성 유지 기법으로 접근 되어야 한다. 즉, 이들 페이지를 읽기 위해 버퍼에 장착하는 경우 그 페이지가 클러스터 내에서 최신 버전임을 버퍼 관리기가 보장하는 방식이다.

파일의 경우 별도의 잠금을 S2PL 을 위해 사용하며 이 잠금의 단위에 따라 적용 가능한 일관성 유지 기법을 달리하여 성능을 높일 수 있다.

파일 잠금을 통해 접근 하는 경우 그 파일에 속한 페이지는 모두 상기한 바와 같은 회피 기반 일관성 유지 기법을 사용한다.

그러나 레코드 잠금을 통해 접근 하는 경우 검출 기반 일관성 유지 기법을 사용하여 현재 적재 되어 있는 페이지를 최대한 활용할 수 있다. 이는 하나의 페이지에 여러 개의 레코드가 저장되어 있으며 특정 레코드가 다른 노드에서 변경 되어 현재 노드가 적재하고 있는 버퍼의 버전이 비록 최신이 아니더라도 변경 되지 않은 나머지 레코드들에 대한 접근은 적재 되어 있는 버전으로도 충분하다는 점에 착안한 기법이다. 이를 위해 모든 레코드 잠금 테이블에 그 레코드를 접근하기 위해 필요한 최소한의 페이지 버전 번호를 관리하며 레코드 잠금을 승인할 때 그 버전 번호를 넘겨 준다. 이 버전 번호는 버퍼 장착 시 인자로 전달되어 지며 버퍼 관리기는 현재 자신이 장착하고 있을지도 모르는 페이지가 충분한 버전인지를 검출할 수 있다.

파일에 대해서는 이와 같이 잠금 단위에 따라 버퍼 장착 방식이 다를 수 있기 때문에 버퍼 관리기는 두 가지 일관성 유지 기법이 상호 연동 되도록 확장한다.

3.4. 잠금 관리 확장

잠금 관리기는 상기한 바와 같은 이유로 전역 잠금 관리기와 지역 잠금 관리기로 구성한다. 지역 잠금 관리기는 각 노드마다 운영 되며 지역 잠금 테이블을 유지하여 그 노드 내 모든 잠금 요청을 처리한다. 전역 잠금 관리기는 특정한 노드에서 운영되며 전역 잠금 테이블을 유지하여 모든 지역 잠금 관리기에 캐쉬 되어 있는 잠금 권한을 관리하고 중개한다. 전역 잠금 관리기는 필요에 의해 어떤 노드에서도 수행 될 수 있다.

잠금 캐쉬 지원은 잠금 권한(lock authority)에 기반 하여 이루어진다. 잠금 권한은 그 노드가 처리할 수 있는 잠금의 수준을 의미하며 공유 모드 잠금을 처리할 수 있는 공유 모드 권한과 공유 및 배타 모드 잠금을 처리할 수 있는 배타 모드 권한이 있다. 요청된 잠금을 처리할 수 있는 충분한 권한을 지역 잠금 관리기가 소유하고 있다면 전역 잠금 관리기와 교신 없이 그 노드에서 잠금을 승인할 수 있다. 그렇지 않다면 전역 잠금 관리기에 권한을 요청한다. 권한 요청 처리 시 전역 잠금 관리기는 그 잠금에 대한 호환 되지 않는 권한을 다른 노드가 이미 획득하고 있으면 그 노드에 그 권한의 반납을 지시한다. 반납을 지시 받은 지역 잠금 관리기는 해당 노드의 트랜잭션들이 그 잠금 자원에 대해 이미 획득 한 잠금을 반납하기를 기다린 후 그 권한을 반납한다. 권한을 획득한 지역 잠금 관리기는 최종적으로 잠금을 승인하며, 한번 획득된 권한은 전역 잠금 관리기가 반납을 요청하기 전 까지는 계속 그 노드에 캐쉬 된다.

3.5. 버퍼 관리 확장

버퍼 관리기는 3.3 절에서 기술한 바와 같이 두 가지 버퍼 일관성 유지 기법을 지원하도록 확장한다. 이를 위하여 버퍼 관리기는 전역 잠금 관리기가 제공하는 버퍼 잠금을 활용한다.

버퍼 잠금은 버퍼 관리기가 특정 페이지를 적재할 수 있는 권한을 획득하기 위해 사용한다. 버퍼 관리기는 특정 페이지를 특정 모드로 적재하기 위해 그 페이지를 적재하기에 충분한 권한을 가지고 있어야 한다. 만약 권한이 충분하지

않다면 버퍼 잠금을 전역 잠금 관리기에 요청하여 권한을 이양 받아야 한다.

버퍼 잠금 모드는 선택된 일관성 유지 기법과 접근 모드(공유 또는 배타)에 따라 결정 된다. 상호 호환 되지 않는 버퍼 잠금이 새로운 노드에서 요청될 경우 전역 잠금 관리기의 중재에 의해 현재 권한을 소유한 노드가 요청한 노드에게 권한을 이전 한다.

버퍼 관리기는 전역 잠금 관리기에게 적재 권한을 요청하면서 현재 최신 버전을 적재하고 있는 노드에게 그 페이지의 전송을 요구할 수 있으며 전역 잠금 관리기는 이를 처리하는 과정에서 현재 권한을 가지고 있는 노드에게 페이지 전송을 지시할 수 있다.

노드간 페이지 전송 방식은 디스크 경유, 네트워크 경유를 선택할 수 있다.

3.6. 고장 회복 기법 확장

확장된 고장 회복 절차는 고장의 유형에 따라 세 종류의 고장 회복 절차로 분류하였으며 각각의 경우 회복에 소요되는 비용 및 고장의 파급 범위가 다르다. 첫째, 전역 잠금 관리기만 고장 난 경우이다. 이 경우에는 단순히 각 지역 잠금 관리기의 정보를 재구성하는 것만으로 회복이 완료된다. 이러한 유형은 전역 잠금 관리기를 별도의 노드에서 운영하는 중 그 노드에 고장이 발생하는 경우 발생한다. 회복 시간이 가장 빠르며 철회되어야 하는 트랜잭션도 없어 고장의 파급이 가장 작다. 둘째, 서비스 노드에 고장이 발생한 경우이다. 이 경우에는 그 노드가 적재하고 있던 버퍼의 내용을 로그를 이용하여 모두 복구한 후 그 노드에서 수행 중이던 트랜잭션을 모두 철회하기 위해 로그 기반 재시작 회복을 수행한다. 회복 비용은 증가 하지만 이 과정이 진행 중이라도 고장이 발생하지 않는 노드는 상당 부분 고장과 관련 없이 진행이 가능하여 고장의 파급 효과가 어느 정도 제한 된다. 즉, 정상 노드가 고장과 관련 없는 자원을 접근한다면 정상적인 운영이 가능하다. 셋째, 전역 잠금 관리기와 하나 이상의 서비스 노드에 고장이 발생한 경우이다. 이 경우에는 먼저 현재 정상인 노드의 지역 잠금 관리기의 정보를 재구성하여 전역 잠금 관리기를 기동 시킨 후 고장 난 서비스 노드에 대해 로그 기반 회복을 수행한다. 이 경우에는 회복 비용이 상대적으로 크며 회복의 파급도 매우 커서 정상인 노드에서 극히 제한적으로 정상 작동이 가능하다. 즉, 정상인 노드가 접근하려는 자원이 이미 그 노드가 획득하고 있는 자원이어서 전역 잠금 관리기 등과 교신 없이 수행할 수 있을 경우에 한해 정상적인 수행이 가능하다.

로그기반 회복을 위해 각 노드는 전용의 로그 장치를 유지한다. 특정 노드에서 발생하는 모든 로그 레코드는 그 노드에 할당된 로그 장치에 기록된다. 어떤 노드이건 사용중인 로그 장치가 가득 차면 빈 로그 장치 풀(empty log pool)에서 새로운 로그 장치를 할당 받는다.

WAL(write ahead logging)은 노드간 페이지 전송 시에도 보장 될 수 있도록 확장 된다. 기존의 WAL 은 특정 더티(dirty) 페이지가 디스크로 기록 되기 전 관련 로그가 로그 장치에 먼저 쓰여 지기를 보장해야 하는데 이는 더티 페이지 전송 시에도 관련 로그를 기록 시키는 것으로 확장 한다.

모든 노드의 로그 장치는 어떤 노드에서라도 접근 가능하도록 공유 디스크상에 유지한다. 그 이유는 첫째, 특정 노드에 하드웨어적인 고장이 발생하여 그 노드가 가용하지 않을 경우 현재 정상인 노드 중 하나가 대신 로그 기반 회복을 수행해야 할 수 있기 때문이다. 둘째, 노드간 페이지 교환 시 디스크 기록을 선택하지 않았다면, 고장 난 노드에 장착되어 있던 더티 버퍼의 내용이 다른 노드에서 발생한 변경들을 누적하고 있을 수 있으며 이들을 복구하기 위해서 그 페이지를 변경한 모든 노드의 로그를 병합하여야 회복이 가능하기 때문이다.

3.7. 고장 이전에 의한 고가용성 지원

고장 이전은 고장의 탐지 단계와 고장 회복 절차 기동 단계로 구성된다. 고장이 어떠한 방법으로든 탐지 되었을 경우 3.6 절에서 기술한 바와 같은 고장 회복 절차가 기동 되어야 한다.

고가용성을 위한 바다-IV 의 확장에서는 자체적인 고장 탐지 및 고장 회복 절차의 자동적인 기동 기능은 포함하지 않는다. 이 기능은 바다-IV 와 운영 체제 사이에서 바다-IV 와 독립적인 클러스터 관리 도구가 수행하도록 확장 한다.

이를 위하여 바다-IV 확장과 별도로 클러스터 관리기와 노드 관리기로 구성되는 클러스터 관리 도구를 구현한다. 모든 노드에는 노드의 상태를 감시하는 노드 관리기가 운영되며 주기적으로 그 노드의 상태 및 확장된 바다-IV 의 운영 상태를 클러스터 관리기에게 보고한다. 클러스터 관리기는 특정 고장이 보고 되었을 경우 3.6 절에서 기술한 바와 같은 고장의 유형을 판별하고 고장 회복을 수행할 노드를 선정하고 최종적으로 확장된 바다-IV 가 제공하는 고장 회복 절차를 기동하는 등의 일련의 고장 이전(fail-over) 절차를 지시한다.

이와 같은 구조로 인해 바다-IV 의 확장과 클러스터 관리 도구의 역할이 분명하게 분리 되어 질 수 있으며 또한 클러스터 관리 도구의 지속적인 확장 또는 새로운 클러스터 관리 도구의 채용 등이 확장된 바다-IV 와 별개로 진행 될 수 있는 이점을 가진다.

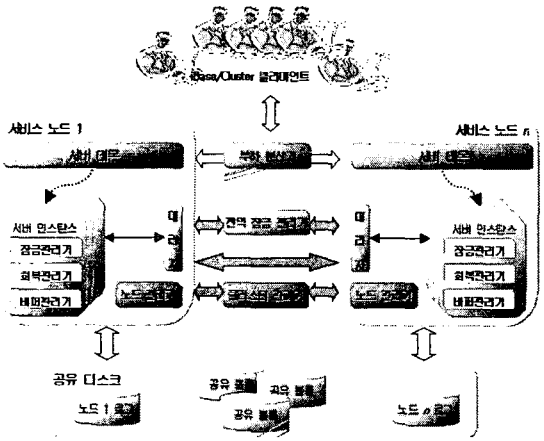
4. iBASE/Cluster 의 구현

4.1. 시스템 구조

상기한 바와 같은 내용은 iBASE/Cluster 로 구현하였다. iBASE/Cluster 의 구현 환경은 다음과 같다.

노드 개수	4
노드	Compaq Server 1.2GHz, 512M RAM, 1 CPU QLogic Fiber Channel Host Bus Adapter Gigabit Ethernet Adapter Fast Ethernet Adapter
운영체제	RedHat Linux V7.1, LVS V1.17
저장장치	Eurologic RAID controller with 128GByte
저장장치망	Brocade Fiber Channel Switch
클러스터망	3COM Gigabit Ethernet Switch
서비스망	3COM Fast Ethernet 100Mbps

구현된 iBASE/Cluster 의 시스템 구조는 그림 1 과 같다. 총 네대의 서비스 노드로 구성 되었으며 각 노드에는 iBASE/Cluster 서버 데몬, 대리자 프로세스, 노드 관리기 등이 상주한다. 부하 분산기, 전역 잠금 관리기, 클러스터 관리기 등은 서비스 노드 중 하나에서 운영 된다.(별개의 독립 노드에서의 운영도 가능하다) 모든 노드는 빠른 동기화를 위해 Gigabit Ethernet 으로 전용의 클러스터망을 구축 하였으며 공유 디스크 접근을 위해 FC 기반 저장장치망을 사용하였다. 각 노드는 공유 디스크상에 자신을 위한 로그 장치를 가지며 어떤 노드에서든 접근할 수 있다. 데이터베이스는 모든 노드가 공유한다.



[그림 1] iBASE/Cluster 시스템 구조

4.2. 동작 시나리오

iBASE/Cluster 의 기동 시나리오는 다음과 같다. 부하 분산기, 전역 잠금 관리기, 클러스터 관리기를 먼저 기동한다. 이들은 서비스 노드와 별개의 노드에서 운영 되거나 서비스 노드 중 하나에서 운영될 수 있다. 이들이 기동 된 후 n 개의 서비스 노드에 iBASE/Cluster 서버 데몬과 대리자 프로세스 그리고 노드 관리기가 수행된다. 대리자 프로세스와 노드 관리기가 기동 직후 클러스터 내의 전역 잠금 관리기의 위치 및 클러스터 관리기의 위치를 각각 파악하여 이에 접속을 하면 서비스 준비가 완료된다.

클라이언트의 접속 시나리오는 다음과 같다. 클라이언트가 LVS 상의 가상 IP 를 이용하여 클러스터에 접속하면 이를 부하 분산기가 적절한 스케줄링에 의해 특정 노드의 서버 데몬으로 접속을 분배 시킨다. 서버 데몬은 클라이언트 접속 시 서버 인스탄스를 기동(fork) 시킨다. 기동 된 서버 인스탄스는 그 노드의 대리자와 연결되고 기동이 완료 된다.

기동 완료된 서버 인스탄스는 대리자, 전역 잠금 관리기와 교신 하며 클라이언트의 질의 처리 요구를 처리한다. 전역 잠금 요청 및 버퍼 요청은 대리자에 의해 전역 잠금 관리기로 중계되며, 전역 잠금 관리기는 잠금 및 버퍼 회수가 필요할 겨우 이를 다시 다른 노드의 지역 잠금 관리기 및 버퍼 관리기로 전달할 수 있다.

고장 발생시의 시나리오는 다음과 같다. 노드 1 에 고장이 발생한 경우 클러스터 관리기는 이를 감지하고 이를 분석하여 노드 1 을 즉시 격리 시키고 적절한 유휴 노드(예를 들어 노드 2)에게 노드 1 의 회복을 지시한다. 노드 2 에는 이 지시에 의해 회복 프로세스가 기동 되며 이 프로세스는 모든 노드의 로그를 가상 병합하여 노드 1 의 고장을 회복한다.

5. 결론

본 논문에서는 공유 디스크 클러스터 환경에서 운영되는 DBMS 개발을 위해 바다-IV 를 기반으로 개발된 iBASE/Cluster 의 확장에 대한 연구 결과를 다루었다. 본 논문에서는 먼저 바다-IV 확장을 위해 고려되어진 프로세스 구조, 전역 동시성 제어, 버퍼 일관성 제어, 전역 회복, 고장 이전(fail-over) 지원 기능 등에 대해 살펴 보고 이를 기반으로 구현된 iBASE/Cluster 시스템 구조 및 운영 시나리오를 다루었다.

향후 연구 방향으로 구현된 시스템에 대한 튜닝, 성능 평가 등을 통한 안정화, 클러스터를 위한 트랜잭션 모델의 확장, 병렬 질의 처리 등에 대한 연구가 계획 중이다.

참고문헌

[1] 김진미, 온기원, 김학영, 지동해 , "클러스터링 컴퓨팅기술," 1999

[2] 최재영, 황석찬, "클러스터를 위한 소프트웨어 도구,"정보과학회지, 제 18 권 3 호, pp40~47.

[3] Zhang, W. "Linux Virtual Servers for Scalable Network Services," http://www.LinuxVirtualServer.org/

[4] Unknown, "Sun Cluster 3 Architecture, A Technical Overview," http://www.sun.com/clusters, Sun Microsystems, Inc., Dec., 2000

[5] S. R. Soltis, T.M. Ruwart, and M.T. O'keefe,"The Global File Systems," Proc. Of the 5th NASA Goddard Conference on Mass Storage Systems and Technologies, 1996

[6] Sang G. Oh, and Jang S. Lee, "SANiqueTM:A SAN File System for Linux Cluster," Technical White Paper – Draft, MacroImpact Co. Ltd., 2001

[7] 김경배,김영호,김창수,신범주, "SAN 을 위한 전역 파일 공유 시스템의 개발," 2001 년 3 월 정보과학회지 제 19 권 제 3 호 통권 제 142 호 2001 년 3 월 ISSN 1229-6821

[8] Mark Bauer,"Oracle8i Parallel Server Concepts, Release 2(8.1.6)," Part No. A76968-01, Oracle Corporation, Dec., 1999

[9] "DB2's Use of the Coupling Facility for Data Sharing," IBM System Journal Vol. 36, No. 2, 1997

[10] Sohan DeMel, "Oracle9i/Parallel Server:Cache Fusion Delivers Scalability," Oracle White Paper, Oct., 2000

[11] C. Mohan, Inderpal Narang,"Recovery and Coherency-Control Protocols for Fast Intersystem Page Transfer and Fine-Granularity Locking in a Shared Disks Transaction Environment," IBM Research Report RJ8017, March 15,1991

[12] Rahm, E., "Concurrency and Coherency Control in database sharing systems," Technical Report 3/91, University of Kaiserslautern, Dec., 1991

[13] Rahm, E. "Recovery Concepts for Data Sharing Systems," Proc. of the 21st Int. Symposium on Fault-Tolerant Computing, Montreal, IEEE Computer Society Press, pp. 368-375, 1989

[14] 이미영, 채미옥, 박순영, 이규웅, 김완석, 김명준, "멀티미디어 통합 정보 검색 프레임워크를 지원하는 OODBMS," 한국정보처리학회 논문지, 7 권, 2S 호, 2000