Journal of Korea Multimedia Society Vol. 7, No. 4, April 2004(pp. 505-517)

고가용성 클러스터 파일 시스템 SANique^{TM}의 분할그룹 탐지 및 회복 기법

이 규 웅†

요 약

본 논문은 저장 장치 전용 네트워크인 SAN에 직접 연결된 저장장치들을 특정한 서버의 도움 없이 접근할 수 있는 클러스터 파일 시스템 SANique^{TM}의 설계방법을 기술하며, 특히 회복 관리기의 기능 및 특징을 설명하고, 이를 위한 시스템 구성요소 및 오류 탐지 절차를 기술한다. 또한 클러스터 내의 오류 발생 상황 중에서 탐지하기 어려운 분할 그룹 문제를 정의하고 이 문제를 해결하기 위하여 SANique^{TM}의 회복 관리기에서 사용한 공유 디스크를 활용한 오류 탐지 및 회복 방법을 제안한다.

Recovery Management of Split-Brain Group in Highly Available Cluster File System SANique^{TM}

Lee, Kyu Woong†

ABSTRACT

This paper overviews the design details of the cluster file system SANique^{TM} on the SAN environment. SANique^{TM} has the capability of transferring user data from shared SAN disk to client application without control of centralized file server. We, especially, focus on the characteristics and functions of recovery manager CRM of SANique^{TM}. The process component for failure detection and its overall procedure are described. We define the split-brain problem that cannot be easily detected in cluster file systems and also propose the recovery management method based on SAN disk in order to detect and solve the split-brain situation.

Key words: recovery(회복), failure(오류), split-brain(분할그룹), cluster system(클러스터 시스템), file system(파일 시스템)

1. 서 론

웹 서비스의 수요 증가와 네트워크의 발전으로 인해 인터넷 기반 응용 시스템들이 요구하는 데이터의 종류가 다양해지면서 멀티미디어 데이터를 포함한 대용량의 데이터 처리 요구가 급증하고 있다. 다수의 인터넷 사용자에게 웹을 통한 대용량의 멀티미디어 데이터 서비스를 효율적으로 제공해주기 위해서 서버 시스템내의 저장장치 구조의 변화가 필요하다. 즉 동시 접근을 요구하는 사용자가 많아지고, 각각의 접근 요구마다 데이터의 크기가 커지면서 저장장치의 부하가 증가하여 대용량의 멀티미디어 데이터를 서비스 해주는데 어려움이 증가하고 있다. 기존의 클라이언트/서버 구조에 기반한 서버 지향적 파일 시스템은 서버 자체적인 용량의 한계와 서버와 클라이언트 간의 잦은 데이터 전송에 따른 메모리 복사가 요구되며, 서버의 오류 발생시 모든 클라이언트들이 서비스 받지 못하는 치명적인 오류를 유발할 수 있게 된다.

※ 교신저자(Corresponding Author) : 이규웅, 주소 : 강원도 원주시 우산동 660(220-702), 전화 : 033)730-0488, FAX : 033)730-0480, E-mail : leekw@sangji.ac.kr
접수일 : 2003년 6월 27일, 완료일 : 2003년 10월 6일
† 정회원, 상지대학교 컴퓨터정보공학부 조교수
※ 본 연구는 2001년도 상지대학교 연구비 지원에 의한 것임.

네트워크 부착형 저장장치(Network Attached Storage; NAS)와 저장장치 전용 네트워크(Storage Area Network; SAN)는 서버에 종속적으로 연결되던 저장장치들을 화이버 채널을 이용한 고속의 전용 네트워크에 직접 연결하여 특정 서버의 제어 없이 네트워크를 통해 직접적인 접근이 가능한 데이터 중심적인 새로운 저장장치 환경이다. 현재 SAN을 통한 저장장치의 클러스터링에 대한 연구가 증가하고 있으며, 이러한 환경에서의 클러스터 파일 시스템 및 시스템 소프트웨어들이 상용화되고 있다[4,6,7].

SAN 기반 클러스터 파일 시스템은 분산 파일 시스템의 기능을 모두 지원하며 또한 그림 1에 나타낸 바와 같이 SAN에 직접 연결된 저장장치들을 특정한 서버의 도움 없이 전용 네트워크를 통해 접근할 수 있으므로 기존 분산 파일 시스템보다 확장성이나 가용성이 우수하다[1]. 기존 분산 파일 시스템들이 하나의 중앙 집중적인 서버에 의한 전체 저장장치 관리라는 서버 시스템의 병목현상 단점을 가지고 있는 반면, SAN 기반 클러스터 파일 시스템은 특정한 서버와의 접근없이 SAN에 부착된 공유 저장장치들을 자유롭게 접근할 수 있다는 장점을 제공한다. 그러나 클러스터내의 각 서버들은 파일 공유를 위해서 기존의 파일 시스템을 사용할 수 없으며, 클러스터를 위한 전용 파일 시스템을 운영해야 한다. 즉 파일 공유를 위한 동시성 제어나 메타 데이터의 관리기능을 할 수 있는 클러스터 파일 시스템으로 대치되어야 한다. SANique^{TM} 파일 시스템은 이와 같은 환경에서 사용할 수 있는 클러스터 파일 시스템이다.

클러스터 파일 시스템은 그림 1과 같이 여러 서버들이 클러스터에 참여하여 하나의 파일 시스템을 구성하게 된다. 기존의 클라이언트/서버 구조가 아니므로 디스크 접근시 특정 서버와의 연결이 필요없지만, 파일의 공유시 필요한 동시성 제어나 메타 데이터의 관리를 위해서 클러스터 파일 시스템의 일부 참여 노드들이 서로 송수신을 할 필요가 있다. 예를 들어 그림 1의 노드 1이 SAN 디스크 1의 특정 파일에 기록을 수행할 때, 동시에 노드 2에서 같은 파일에 대한 접근을 시도하는 경우, 그 파일의 메타 데이터 일관성을 위해 동시성 제어를 할 필요가 있으며, 파일에 대한 메타 데이터 정보 변경을 구성 노드들이 알 수 있도록 해야 한다. 그러기 위해서 노드 1은 관련된 다른 노드들과 메타 데이터 송수신을 해야 한다. 클러스터 파일 시스템 서비스중 구성 노드 하나 또는 다수의 노드가 예기치 않은 오류로 인해 시스템 고장을 일으키거나, 범용 네트워크의 오류로 인해 메타 데이터 정보 송수신이 불가능한 경우, 고장 노드를 클러스터 내에서 제거하고 나머지 참여노드들로 구성되는 클러스터 파일 시스템 환경을 운영할 수 있어야 한다. 즉, 오류가 발생한 노드가 어떤 노드이며 그 오류 노드의 역할을 클러스터 내의 다른 노드가 대신(failover)하여 수행할 수 있도록 오류 탐지 및 회복 기능이 지원되어야 한다.

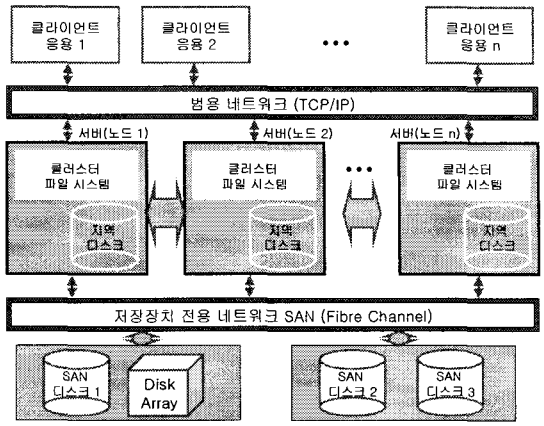


그림 1. SAN 기반 클러스터 파일 시스템의 구성도

더구나 범용 네트워크 오류로 인해 참여노드들이 서로 메타 데이터를 송수신 할 수 없는 두 개 또는 여러 개의 그룹(split-brain group)으로 분할되는 경우에는 어떤 노드가 오류 노드인지 판별하기 힘든 상황이 된다. 이와 같은 상황에서 분할된 각 그룹은 서로 상대 그룹에 해당하는 노드들이 모두 오류상태인 것으로 판단하게 되어 상대 그룹 노드의 역할을 각각의 그룹 내에서 떠맡게 된다. 이러한 현상으로 인해 클러스터 공유 파일 시스템의 일관성(consistency)이 위반되게 되어 존재하지 않는 데이터를 접근하려는 시도가 발생할 수 있게 된다.

따라서, 본 논문은 이와 같은 클러스터 파일 시스템내의 오류로 인한 분할그룹 문제에 대하여 오류가 발생한 노드들을 정확하게 판별할 수 있는 방법을 제시하며, 제시한 탐지 기법을 기반으로 클러스터 파일 시스템의 회복기법에 대하여 설명한다.

본 논문의 구성은 다음과 같다. 2장에서 기존 클러스터 파일 시스템의 종류와 기능에 대하여 분석하고, 본 논문의 기반 시스템이 되는 SANique^{TM} 시스템의

구성요소에 대하여 설명한다. 3장에서 SANique™ 시스템의 오류 탐지 절차 및 이를 위한 시스템 프로세스 구성도를 보이며 또한 클러스터 파일 시스템의 오류 및 분할그룹에 대한 정의를 내리고, 클러스터 파일 시스템 내에서 발생할 수 있는 분할그룹의 상황에 대해 분석한다. 여러 분할그룹 상황을 기반으로 문제점을 파악하여 이를 해결할 수 있는 분할그룹 탐지 및 회복 기법을 4장에서 제시한다. 끝으로 5장에서 결론을 맺는다.

2. 기존 연구 및 SANique™의 구성

2.1 클러스터 파일 시스템의 특성 및 기존 연구

네트워크의 발전으로 인하여 지역적 파일 시스템은 점차 서버들의 클러스터를 통한 분산 파일 시스템(distributed file system) 또는 공유 파일 시스템(shared file system)으로 발전되고 있다. 다수의 원격 노드에 존재하는 클라이언트로부터 공유 저장 장치가 동시에 마운트(mount)될 수 있으며, 동시에 원격지의 클라이언트들에 의해 동시 접근이 가능한 파일 시스템이다.

전형적 클라이언트-서버 파일 시스템은 서버에 의해 파일 식별 공간, 파일 접근 허가 권한 등을 제공받으며, 파일 이름과 파일 오프셋을 디스크 블록 어드레스로 매핑하는 기능을 제공받고 있다. 이와 같은 클라이언트-서버 분산 파일 시스템은 각 클라이언트에서 사용되는 지역 파일 시스템 명령어에 의해 분산 저장된 파일을 접근할 수 있는 장점을 갖고 있다. 또한 RPC, XDR, TCP/IP 등과 같은 표준 네트워크 프로토콜에 의해 각 클라이언트 시스템의 네트워크 하드웨어와 프로토콜에 대한 독립성을 제공받을 수 있다. 이와 같은 구조는 서버 노드의 처리능력에 종속적인 제한된 확장성과 가용성을 제공한다는 단점을 갖게 된다. 이러한 분산 파일 시스템의 대표적인 예는 썬 마이크로 시스템사의 NFS(Network File System)과 AT&T사의 RFS(Remote File Sharing System), UCB 대학의 xFS(Andrew File Systems) 등이 있다[2,8,10].

다른 종류의 파일 공유 방법은 미들웨어 방식의 공유로서, 파일 공유를 위해 운영체제를 수정하지 않고 응용 프로그램과 운영체제 사이의 파일 관련 요구사항들을 가로채어 파일 공유가 가능하도록 하는 모듈을 이용하는 방식이다. 이와 같은 기능을 제공하는 제품으로는 IBM Tivoli사의 SANergy를 들 수 있다. 이 제품에서는 파일 공유의 동시성을 제어하기 위하여 서버 역할을 하는 메타 데이터 제어기 노드를 별도로 운영하여야 한다.

궁극적으로 클러스터링 되어 있는 서버들 간의 데이터를 공유하기 위해서는 데이터 접근시 어떠한 중앙집중적 서버의 제어도 받지 않아야 되며, 서버의 오류로 인해 전체 클러스터 내의 노드들이 데이터 서비스가 중단되어서도 안된다. IBM의 Tivoli사에서 나온 SANergy는 클러스터 간의 파일 공유 솔루션으로 출시되었으며 실용 업무에 사용되고 있다. 그러나, 파일 공유의 특성상 대용량 멀티미디어 데이터를 취급하는 환경이 많은데 비해, 공유 데이터 전송에 많은 시간이 소요된다는 단점을 지니고 있다. 또한 서버 역할을 하는 MDC 모듈과 클라이언트 모듈로 구성되는 클라이언트/서버 형식을 취하고 있어, 서버에 오버헤드가 발생 할 수 있으며 병목현상의 문제점을 안고 있다. 또한 클러스터내의 노드 오류로 인한 전체 시스템 서비스의 중단과 이를 처리하는 장애복구 시간이 길어 고가용성을 제공하지 못한다

Minnesota 대학의 GFS(Global File System)은 주로 개인적, 연구용 목적으로 사용되었으나 Sistina라는 기업에서 인수, 개발하여 현재 제품으로 출시되고 있다[3,5,6]. 이 공유 파일 시스템은 GFS 로크 서버 역할을 하는 노드와 클라이언트 노드들로 구성되는 비대칭형 구조이다. 각 클라이언트들은 GFS 로크 서버의 제어 하에 로크를 획득한 후, 공유 디스크를 접근하게 된다. 이 제품 역시 비대칭형 구조의 전형적인 문제인 병목현상으로 인한 성능 저하 현상을 갖게 되며, 로크 서버의 오류로 인하여 전체 시스템 서비스가 중단되는 저가용성 문제를 갖는다.

최근에 출시된 Matrix 서버 제품은 폴리서브라는 회사의 제품으로 본래 목적은 클러스터링 솔루션으로 개발되었으나, 최근 공유 파일 시스템의 수요가 증가함에 따라 공유 파일 기능이 추가된 경우이다. 최근 출시 제품이므로, 아직 시장 평가가 많이 이루어 지지 않은 상태로서 현재 10노드 클러스터 규모를 지원하며 플랫폼의 운영체제 또한 리눅스로 제한되고 있다.

썬 마이크로 시스템의 기술진들이 설립한 Veritas에서 출시한 SAN Point Foundation Suite는 클러스터 파일 시스템, 볼륨 관리기를 포함하는 제품으로

가장 최근 출시되었으며, 아직 오류로 인한 고가용성 제공의 미흡과 비용 및 번들 구성으로 인한 사용자 요구 만족등의 문제점을 안고 있다[4].

2.2 SANique^{TM} 시스템 구조

SAN 기반 클러스터 파일 시스템이란 SAN에 직접적으로 연결된 공유 디스크들을 클러스터를 구성하는 다수의 노드들이 서로 동시에 접근할 수 있도록 해주는 분산 파일 시스템의 한 일종이다. 따라서 다수 노드의 접근에 의한 메타 데이터의 일관성 및 무결성을 보장해 주어야 한다는 목적은 기존 서버 지향적인 분산 파일 시스템과 같다. 그러나 기존 분산 파일 시스템에서 발생하는 서버 병목현상을 제거하여 효율적인 디스크 접근 성능을 제공하는 클러스터 파일 시스템이다. 즉, 기존 시스템은 공유할 모든 디스크들을 한 대의 서버에 의해서 관리되도록 설계되어 있으므로, 서버에 대한 병목현상을 피할 수 없게 된다. 또한 중앙집중적인 관리 서버의 오류로 인해 전체 파일 시스템의 서비스가 중단될 수 있다. 클러스터 파일 시스템은 특정한 서버에 의해 공유 디스크가 관리되는 것이 아니라, 클러스터 내의 모든 노드들이 공유 디스크를 서버와의 연결없이 자유롭게 직접 접근할 수 있는 파일 시스템이다.

SANique^{TM}은 클러스터 서버들을 위한 저장 장치 솔루션으로서, 전형적인 분산 파일 시스템의 병목현상을 제거하였으며, I/O 대역폭을 늘려 성능을 최대화 한 클러스터 파일 시스템을 제공한다. SANique^{TM}은 그림 2와 같이 클러스터 파일 시스템 지원을 위해, 볼륨 관리기(cluster volume manager), 로크 관리기(cluster lock manager), 버퍼 관리기(cluster buffer manager), 회복 관리기(cluster recovery manager) 등으로 구성된다.

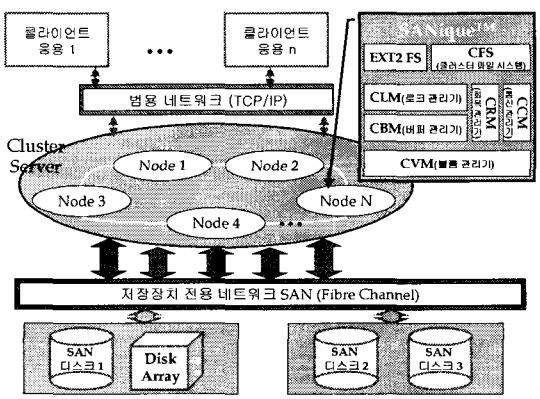


그림 2. SANique^{TM}의 시스템 구성도

클러스터 파일 시스템 CFS

SAN 상에 연결된 모든 단일 노드들은 클러스터 파일 시스템인 SANique^{TM}의 CFS를 통하여 SAN에 직접적으로 부착된 디스크들에 대해 병행적 공유 접근이 가능하다. SANique^{TM}의 CFS는 64비트 주소 공간을 갖는 저널링(journaling) 파일 시스템이다. 파일의 크기에 제한을 두지 않으므로, 최대 2^{64}바이트 크기의 단일 파일을 생성할 수 있다. CFS는 수퍼블록 영역, 비트맵 영역, 데이터 블록 영역으로 구성되며, 수퍼블록 영역은 메타 데이터 저장공간으로 활용되고, 비트맵 영역은 디스크 공간의 할당 영역과 가용 영역을 관리하기 위한 공간으로 이용된다. 전형적인 분산 파일 시스템 상에서는 이러한 관리 영역들을 하나의 중앙집중적인 파일 서버가 책임지고 있으므로, 상호 배타적인 접근만이 가능하며 서버에 오류로 인한 치명적인 전체 시스템 서비스의 중단을 유발할 수 도 있다. SANique^{TM} CFS는 비트맵 접근 단위를 세분화하여 다수의 노드에 분산 관리하도록 하여 동시 접근 및 수정이 가능하므로 병렬성을 극대화 할 수 있다.

로크 관리기 CLM

SANique^{TM} 로크 관리기 CLM은 클러스터내의 다수의 노드에 의해 동시 접근 되는 파일에 대해 접근 직렬성을 보장해 준다. 공유 볼륨상의 파일 접근을 직렬화 하고 일관된 데이터 접근을 보장하기 위하여 파일 수준에서의 병행수행 제어 방법을 제공한다. CLM에 의하여 같은 파일에 접근하는 모든 연산은 파일 단위의 데이터 일관성을 보장 받게 된다. 또한 특정 서버를 통한 로크 관리 기능을 제공하게 되면, CLM 서버에 대한 병목현상을 유발하게 되고, 가용성 또한 감소하게 되므로, CLM 기능을 클러스터 내의 모든 노드에 분산하여 수행할 수 있도록 설계하였다.

버퍼 관리기 CBM

SANique^{TM}의 버퍼 관리기 CBM은 클러스터 내의 노드들이 내부 캐쉬에 저장하고 있는 캐쉬 블록들을 다른 노드에서 활용할 수 있도록 전역 버퍼 관리 기법을 사용하고 있다. 요청한 데이터 블록이 자신의 내부 버퍼에도 존재하지 않고, 클러스터 내의 다른 노드에도 모두 존재하지 않는 경우에만 디스크에서

데이터를 읽어오게 하는 상호 협조 캐싱 기법을 적용하여 전역 버퍼를 관리한다. 특히 사용자 데이터 뿐만 아니라 클러스터 파일 시스템을 관리하기 위한 메타 데이터에 대한 캐싱을 상호 협조 캐싱으로 처리하여 파일 시스템 관리를 보다 효율적으로 할 수 있다.

볼륨 관리기 CVM

SANique™의 볼륨 관리기 CVM은 SAN에 직접적으로 부착된 모든 디스크 공간을 공유 가능한 하나 또는 그 이상의 가상 볼륨으로 구성하여 저장 풀(storage pool)을 형성해 주는 기능을 제공한다. 물리적 디스크 집합으로부터 논리적 디스크 영역을 구성하고 이를 파일 시스템의 주소 공간으로 전환할 수 있는 매핑 테이블을 구축하여 클러스터 내의 노드들이 분할 담당하도록 한다. 논리적 볼륨은 물리적 디스크의 구성 방식에 따라 스트라이핑, 미러링(RAID-1), 패러티 스트라이핑(RAID-5)등 다양한 형식으로 지원될 수 있다.

회복 관리기 CRM

SANique™의 회복 관리기 CRM은 클러스터내에 참여하고 있는 노드중의 하나가 오류로 인하여 서비스 제공에 장애를 일으키는 경우, 온라인 상태에서 오류를 탐지 및 회복하여 전체 클러스터 파일 시스템을 정상적으로 서비스할 수 있는 상태로 복구시켜주는 기능을 갖는다. CRM은 커널 수준에서 수행되는 시스템 모듈로서 실제 오류 탐지 및 조사를 위해 사용자 프로세스 수준의 고가용성 서비스 계층과 상호 협조하에 수행된다.

SAN 기반 클러스터 파일 시스템은 두 종류의 네트워크, 즉 범용 네트워크(TCP/IP 기반)와 광채널과 같은 고속 매체를 이용한 저장장치 전용 네트워크(SAN)으로 구성된다. 범용 네트워크는 응용 서비스를 제공하거나 클러스터 파일 시스템 운영을 위한 노드들 간의 메타 데이터 전송을 위하여 사용되며, SAN은 주로 데이터 블록의 전송에 사용된다. 이러한 시스템에서 범용 네트워크에만 문제가 발생하고, SAN은 정상적으로 작동하게 되는 경우, 노드들 간에 메타 데이터 전송이 되지 않아 전체적인 파일 시스템의 불일치 상태를 유발하게 된다. 따라서 이러한 오류를 탐지하는데 있어서 가장 큰 문제는 두 노드가 서로 상대 노드를 오류 노드로 인식하게 되는 분할그룹(split-brain) 문제이다. 본 논문에서는 SANique™ 시스템에 직접 이용하고 있는 분할그룹 상태의 탐지 방법과 회복 방법에 대하여 논의한다.

3. 클러스터 파일 시스템의 오류 및 분할그룹

3.1 SANique™ 오류 탐지 시스템 구성 및 탐지 절차

클러스터 파일 시스템 상에서 발생하는 오류는 프로세스 오류, 시스템 오류, 장치 오류 등으로 구분할 수 있다. 프로세스 오류는 공유 디스크에 대한 데이터 접근 요청을 보낸 클라이언트 측의 프로세스가 비정상적으로 종료하여 발생하는 오류이다. 이러한 오류는 소프트 오류에 분류될 수 있으며, 일반적인 오류 처리 방법에 따라 탐지되고 회복될 수 있다. 시스템 오류란 정전 등에 따른 무전원 상태에 의한 오류, 시스템 하드웨어의 고장으로 인한 오류, 서버측의 시스템 프로세스의 비정상적 종료에 의한 오류 등으로 전체 시스템 서비스를 제공하는데 문제를 발생시키는 오류이다. 또한 장치 오류란 디스크, 네트워크 등의 장애로 인한 하드웨어적인 오류를 말한다. 이러한 단순 장치 오류는 기존의 독립형 시스템에서 적용되던 방법을 그대로 사용하여 처리할 수 있으나, 그 중 네트워크 단절 등으로 인해 클러스터가 두개 이상으로 분할되고, 각 분할된 그룹이 서로 상대그룹을 오류 노드로 인식할 수 있는 문제가 내재되어있어 오류 처리에 문제를 유발하게 된다. 본 절에서는 SANique™ 시스템의 오류 처리 절차를 먼저 설명하고, 후에 분할 그룹 문제의 정의 및 해결방법을 기술하도록 한다.

SANique™ 시스템은 회복 탐지 및 복구를 위해서 서버 프로세스들을 감시하고, 회복 작업을 지시하는 고가용성 서비스 계층을 그림 3과 같이 서버 프로세스의 상위 계층에서 운영한다. 고가용성 서비스 계층의 온라인 서비스 데몬(OLSD)은 주변의 노드에 대한 상태를 점검하여 특정 노드가 온라인 서비스를 제공하고 있는지에 대한 상태 점검과 자신 노드의 서비스 프로세스들이 정상적인 수행을 하고 있는지 자체 검사 작업을 수행한다. 특히, 토큰 기반의 주기적 메시지 전달 방법인 "heartbeat" 기법에 의해 주변의 노드 구성이 이전상태와 달라지는 경우 온라인 서비스 데몬에 이를 통지하고, 온라인 서비스 데몬은 해당 노드의 상태관리 데몬(MCSD)에게 상태관리를 위한 통신을 요청한다. 즉, 주변의 노드 뷰가 달라지

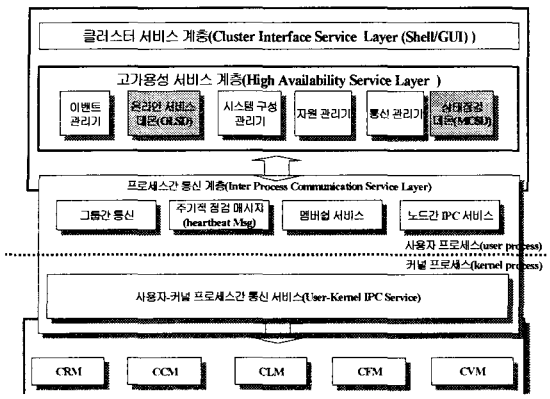


그림 3. SANique^{TM} 시스템의 오류 탐지 데몬 및 상태관리 데몬의 구성

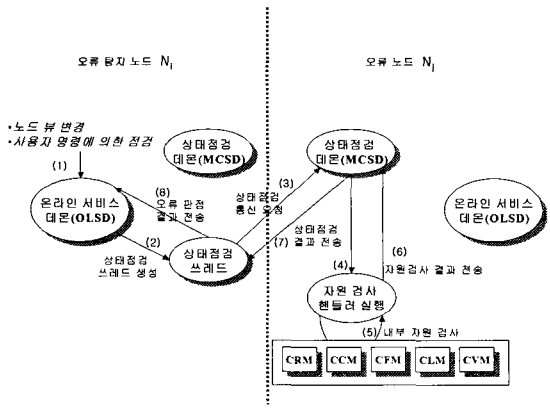


그림 4. 오류 탐지 데몬의 구성 및 수행 절차

게 되면 즉각적으로 온라인 서비스 데몬에 의해 탐지되고 해당 노드의 상태관리 데몬과의 통신을 시도하게 된다. 해당노드의 상태관리 데몬이 응답하지 않는 경우나 또는 그 노드의 일부 자원 즉, 그림 3에 나타난 커널 프로세스 및 시스템 프로세스들에 대한 상태가 올바르지 않은 경우 전체 클러스터 구성에서 그 노드를 제거하고, 그 노드의 기능을 다른 노드에게 전이하는 회복작업을 수행하게 된다.

SANique^{TM} 시스템의 오류 탐지 절차는 그림 4와 같이 오류 탐지 노드 N_i가 오류 예상 노드 N_j에 대해 어떠한 오류가 발생하였는지 상태점검 데몬과의 통신을 통해 이루어지게 된다. 오류 탐지 첫 단계에서는 주기적 메시지에 의한 노드 뷰 변경을 감지하여 온라인 서비스 데몬에 통지하는 단계이며, 둘째 단계는 그림 4의 (2),(3)에 나타난 것처럼 온라인 서비스 데몬에 의해 상태점검 쓰레드를 생성하고, 오류가 예상되는 노드 N_j의 상태점검 데몬과의 통신을 요청하는 단계이다. 세 번째 단계(그림 4의 (4)~(7))는 노드 N_j의 상태점검 데몬이 자신의 노드의 자원, 즉 커널 및 시스템 프로세스들을 점검하고 이를 통지하는 단계이고, 마지막으로 그림 4의 (8)에서와 같이 노드 N_i의 상태점검 쓰레드는 노드 N_j의 오류 여부, 즉 응답이 없는지 또는 응답이 있으나 시스템 자원에 장애가 발생하여 그 노드가 서비스를 할 수 없는지를 판단하여 해당 노드 N_j를 클러스터에서 강제 제거하고 그 기능을 전이받는 회복 작업을 CRM에 지시하는 단계이다. 오류 판단이 완료되면 모든 오류 회복절차는 노드 N_i의 CRM을 통하여 수행하게 된다.

오류 노드 N_j가 전원오류나 완전한 시스템 정지상태인 경우, N_j의 상태점검 데몬은 물론 어떠한 프로세스도 존재하지 않으므로 노드 N_i의 상태점검 쓰레드는 통신 불가로 인해 N_j의 오류를 결정하게 된다. 그러나 노드 N_j의 다른 종류의 오류나 프로세스 오류인 경우 N_j의 상태점검 데몬은 자신의 노드의 자원들을 상태 점검하여 이를 통보한다. 또한 노드 N_j의 상태점검 데몬의 단순 종료로 인해 노드 전체가 오류로 처리되는 것을 방지하기 위해 노드 N_i는 노드 N_j의 상태점검 데몬과 통신에 실패하게 되는 경우 노드 N_j의 온라인 서비스 데몬과 통신을 재시도하게 되어 이중통신을 통한 오류 탐지를 수행한다. 오류 탐지 노드 N_i의 상태점검 쓰레드와 오류 노드 Nj의 상태점검 데몬 및 온라인 서비스 데몬과의 통신 흐름을 도식화하면 그림 5와 같다. 그림 5의 (1)에 해당하는 통신 흐름도는 노드 N_i의 상태점검 쓰레드의 요청에 따라 오류 노드 N_j의 상태점검 데몬이 해당 노드의 모든 자원들, 즉 CFM, CLM 등과 같은 시스템 모듈 및 서버 프로세스들을 점검하여 그 결과를 전송하고 노드 N_i는 노드 N_j의 오류 여부를 판단하게 된다.

그림 5의 통신 흐름도 (2)는 노드 N_j가 완전한 시스템 오류를 발생하여 어떠한 네트워크 연결도 불가능한 상태를 보이고 있다. 그림 5의 통신 흐름도 (3)은 단순한 상태점검 데몬의 오류로 인해 전체 시스템이 오류로 처리되는 것을 방지하기 위하여 노드 N_j의 온라인 서비스 데몬과 재연결을 시도한 후, 상태점검 데몬을 재수행하게 하여 (1) 흐름도와 같은 통신 흐름을 재개하는 통신과정을 보이고 있다.

3.2 분할 그룹 유형 분석 및 문제 정의

클러스터 파일 시스템의 오류 중에서 네트워크에

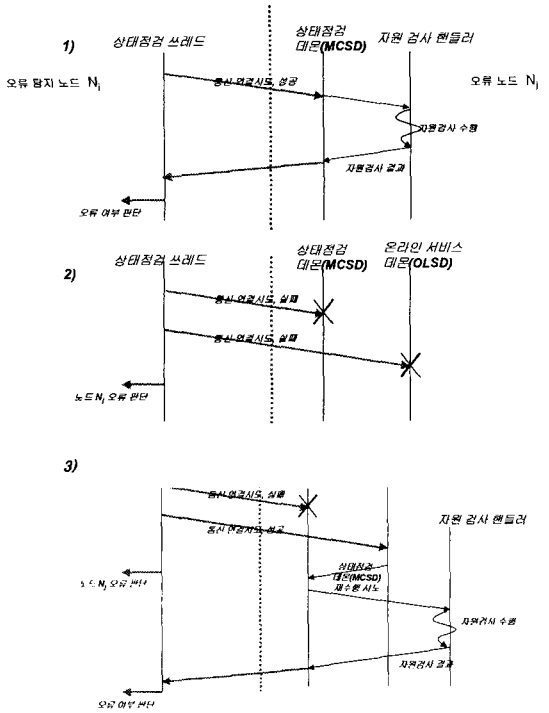


그림 5. 상태점검 쓰레드와 상태점검 데몬 및 온라인 서비스 데몬의 통신 흐름

의한 오류가 발생한 경우에는 각 노드들의 프로세스가 모두 정상적으로 살아있는 경우이므로, 클러스터 내의 노드들이 2개 이상의 그룹으로 분할될 수 있다. 즉, 서로 다른 노드에서 상대방 노드를 시스템 오류 노드로 인지하게 되는 분할그룹 문제(split-brain problem)가 발생하게 된다. 클러스터 파일 시스템과 같이 다수의 노드들이 상호 협조 체제로 수행되는 환경에서, 네트워크 단절로 인해 여러 그룹으로 분할되고 각 그룹이 기능적으로 완벽하지만, 서로 통신이 단절된 상황을 클러스터 환경의 분할그룹이라 정의한다. 그림 6은 두 개의 노드로 구성되는 클러스터 파일 시스템에서 발생하는 분할그룹 문제를 보이고 있다.

노드 1,2는 각각 주기적으로 상대 노드에 점검 메시지를 범용 네트워크를 통해 주기적으로 보내 서로의 상태를 점검한다. 이 때, 저장 장치 전용 네트워크에는 문제가 없으나, 범용 네트워크에 오류가 발생하여 점검 메시지가 전달되지 않게 되면, 각 노드는 서로 상대 노드가 오류 상태인 것으로 판단하게 된다.

만약 실제로 노드 1이나 노드 2가 무전원 상태 등에 의한 파일 시스템 프로세스의 종료로 오류가 발생

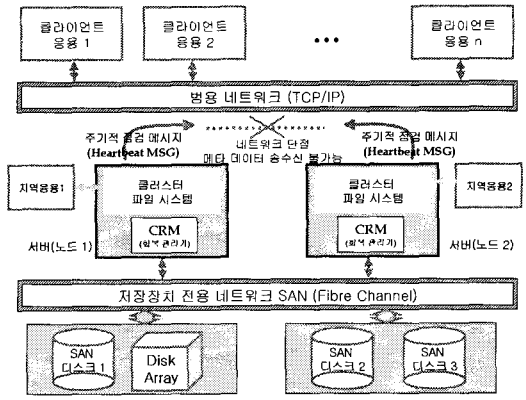


그림 6. 클러스터 파일 시스템의 분할그룹 문제

한 것이라면, 나머지 한 노드에서 오류를 탐지하여 오류 노드의 기능을 모두 떠맡아 전체 시스템 서비스를 재개할 수 있다. 그러나, 그림 6과 같이 단순한 네트워크 단절로 인한 오류 발생 상황인 경우, 노드 1과 2 모두 각각 상대 노드를 오류 처리하여, 상대 노드의 기능을 가져오게 된 후, 시스템 서비스를 양측에서 재개하게 된다. 분할그룹 즉, 노드 1 그룹과 노드 2 그룹은 모두 2개의 파일 시스템 서비스를 동시에 지원하게 된다. 이런 상황에서 두 노드에서 공유 디스크의 같은 파일 F_1에 기록을 위해 접근한다면, 파일 F_1의 메타 데이터 블록(예, inode 블록)을 아무런 수행 제어 없이 변경, 판독하게 되어 전체적인 파일 시스템의 불일치성을 유발하게 된다. 정상적인 시스템 서비스인 경우 각 노드는 파일 F_1을 접근하기 위한 메타 데이터에 대한 로크 정보를 획득하여 서로 직렬 접근을 보장해야 되지만, 이러한 경우 상대 노드가 모두 서비스 정지 상태인 것으로 판단한 후 이므로, 독자적인 수행을 하게 된다. 이러한 문제를 분할그룹 문제라 정의한다.

클러스터 파일 시스템의 분할그룹 문제는 네트워크의 연결 유형에 따라 다양한 형태로 분류될 수 있다. 그림 7은 이러한 다양한 경우의 분할그룹 문제를 세가지 유형으로 분류하여 나타내고 있다.

분할그룹 유형을 분류하기 위하여 오류 그룹과 서비스 그룹을 정의한다.

[정의 1] 오류 그룹

클러스터 파일 시스템내의 노드들로 구성되는 그룹으로서, 그룹 내부 노드들 간의 네트워크 통신이 가능하고, SAN을 통한 공유 디스크와의 통신도 가능한 노드들로 구성되는 그룹이다. 그러나 그룹 외부

로의 네트워크 통신은 불가능하다.

[정의 2] 서비스 그룹

클러스터 파일 시스템내의 노드들로 구성되는 그룹으로서, 그룹 내부 및 외부의 노드들로 모두 네트워크 통신이 가능하며, SAN을 통한 공유 디스크와의 통신도 모두 가능하다. 클러스터 파일 시스템 서비스를 정상적으로 수행하기 위하여 메타 데이터 송수신이 모두 가능한 노드들로 구성되는 그룹이다.

그림 7에 나타낸 바와 같이, 클러스터 파일 시스템의 분할그룹 유형을 다음과 같이 분류한다.

○ 분할그룹 유형 1

이 유형은 네트워크의 일반적인 구성으로 클러스터 내의 노드들이 모두 하나의 네트워크 스위치 장비에 연결되어 있어서 특정 노드의 네트워크 장비 문제로 인하여 특정 노드가 외부로의 네트워크 통신이 단절되는 경우이다. 이런 경우 다수의 오류 그룹이 발생할 수는 있으나 각 오류 그룹의 노드 개수는 반드시 하나이다. 그림 7의 [유형 1]을 보면 두 개의 오류 그룹이 각각 통신 장애로 오류 그룹으로 분류되었으며, 각 그룹의 노드 개수는 하나이다. 클러스터 파일 시스템을 정상적으로 지원할 수 있는 서비스 그룹은 단 하나만이 존재할 수 있다. 서비스 그룹이 두 개 이상 존재하려면, 다중 네트워크 장비를 가진 노드로서 다른 네트워크 장비가 외부 네트워크 스위치 장비와 연결되어야만 한다. 본 연구에서는 이중 네트워크 장비를 가진 노드는 배제하고 유형을 분류한다. 따라서, 분할그룹 유형 1은 서비스 그룹이 반드시 1개 존재하며, 다수의 오류 그룹이 발생할 수 있고, 각 오류 그룹은 하나의 노드만을 갖는다.

그림 7의 [유형 1]인 상황에서는 서비스 그룹과 오류 그룹 1(노드 n-1), 오류 그룹 2(노드 n)이 각각 상대방을 오류로 인식하여 오류 회복 절차를 수행하려고 시도한다. 따라서, 각 그룹에서 오류 회복 절차를 독자적으로 수행하게 되면, 서로 다른 3개의 클러스터 파일 시스템이 공유 파일에 대하여 수행되는 불일치성을 유발할 수 있다. 이 경우, 정상적인 서비스 그룹이 노드 n-1과 노드 n을 클러스터에서 제거해 낼 수 있는 오류 탐지 및 회복 기법이 필요하다.

○ 분할그룹 유형 2

분할그룹 유형 2는 여러개의 네트워크 스위치 장비를 이용하여 클러스터 노드들을 연결한 경우로서, 그림 7의 [유형 2]처럼 네트워크 스위치 장비의 외부

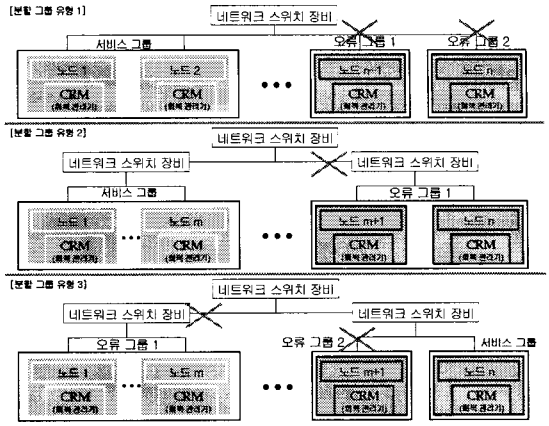


그림 7. 클러스터 파일 시스템의 분할그룹 유형

에서 네트워크 오류가 발생하는 경우, 스위치 내부의 노드, 즉 노드 1부터 노드m 간에는 통신이 가능하지만 외부로는 불가능한 오류 그룹이 발생할 수 있다. 이 유형에서는 [유형 1]과 달리, 각 오류 그룹이 서로 네트워크 통신이 가능한 다수의 노드로 구성될 수 있다. 이 유형도 역시 서비스 그룹은 한 개만이 존재할 수 있다. 따라서 분할그룹 유형 2는 하나의 서비스 그룹과 다수의 오류 그룹이 발생할 수 있으며, 각 오류 그룹은 서로 통신이 가능한 여러 개의 노드로 구성될 수 있다.

그림 7의 [유형 2] 상황에서는 서비스 그룹과 오류 그룹이 상대 그룹을 서로 오류 노드로 인식하게 되는 분할그룹 문제를 발생시킨다. 그러므로 실제 오류를 갖는 오류 그룹을 판별하여 서비스를 중지시키고, 클러스터 내에서 제거할 수 있는 회복 절차를 수행해야 한다.

○ 분할그룹 유형 3

이 유형은 [유형 1]과 [유형 2]의 가능한 모든 조합으로 발생하는 경우로서 한 노드의 네트워크 오류와 네트워크 스위치 장비의 외부로의 오류가 동시에 발생하여 하나의 노드를 갖는 오류 그룹과 다수의 노드를 갖는 오류 그룹, 그리고 정상적인 서비스를 수행할 수 있는 서비스 그룹으로 구성된다. 이러한 상황도 소수의 노드 그룹으로 구성되는 서비스 그룹이라 할지라도 다수의 노드를 갖는 오류 그룹을 적절히 판단하여 클러스터 내에서 제거해야 한다.

제시한 분할그룹 유형을 통하여 분할그룹 문제를 해결하는데 있어서 다음과 같은 문제점을 갖고 있음을 알 수 있다. 첫째, 오류 그룹 자신이 오류 그룹인지

정상 그룹인지를 판별하기 어려우며, 둘째, 클러스터 내의 노드들이 몇 개의 오류 그룹으로 분할되었는지 판단할 수 없다. 셋째, 통신이 되지 않는 노드가 무전원 상태등에 의한 시스템 오류인지, 아니면 단순한 네트워크 오류인지 판단할 수 없다. 마지막으로, 분할그룹 중에서 가장 많은 노드로 구성된 그룹을 선별할 수 없어, 최적의 그룹으로 서비스를 재개하기 어렵다.

위와 같은 문제들은 시스템 오류와 네트워크 오류가 정확히 탐지되지 않아 발생하며, 또한 분할그룹간의 네트워크 통신이 단절되어 각 그룹의 상황을 서로 정확히 인지할 수 없어서 발생하는 문제이다.

4. 공유 디스크 기반의 분할그룹 회복 기법

전형적인 분할그룹 문제를 해결하기 위해 여러 기법들이 제시되었으나, 최적의 해결방법이 아직 제시되지 않았으며, 더구나 클러스터 파일 시스템 상에서의 분할그룹 문제에 대한 구체적인 방법은 아직 연구되지 않고 있다[2,9,12]. 기존의 분산 환경에서는 고가용성을 목적으로 하는 이중화(replication) 기법을 사용하는 클러스터 시스템이 대부분이다. 이중화를 위한 미러링(mirroring) 기법은 다수의 노드가 주 서버와 같은 기능, 같은 데이터를 보관하고 있으므로, 대체 서버를 선택하는 데 있어서 큰 어려움이 없어, 본 연구의 기반 환경이 되는 클러스터 파일 시스템에서 보다 단순하게 해결할 수 있다[8,11,13]. 즉, 분할노드 중에서 어떤 노드가 대체 서버로 서비스를 재개하더라도 전체 시스템 서비스에 큰 변화가 없다. 그러나 클러스터 파일 시스템 환경에서는 그림 7 [유형 3]의 오류 그룹 2가 나머지 모든 노드들을 오류 처리하여 클러스터 내에서 제거한 후, 시스템 서비스를 재개하게 되는 경우 최악의 상태로 파일 시스템 서비스를 재개하게 된다. 이러한 경우 노드 m+1의 지역적 응용들만 파일 시스템 서비스를 받을 수 있으며, 모든 외부 클라이언트 응용들은 서비스 받을 수 없게 되는 극도의 제한적인 서비스를 실행하게 된다. 따라서, 분할된 그룹 중에서 최적의 그룹을 판별하여 다른 모든 그룹을 오류 처리 할 수 있는 기법이 필요하지만, 통신 단절로 인해 분할그룹의 개수와 그룹 내의 노드 정보를 판단할 수 없는 근본적인 문제를 갖고 있다.

4.1 공유 디스크를 활용한 분할그룹 정보 비교 기법

SANique™의 회복 관리기 CRM에서는 분할그룹 문제를 해결하기 위하여 공유 디스크를 활용한다. 즉, 모든 분할그룹들이 서로 범용 네트워크를 통한 통신을 단절된 상태이지만, 저장장치 전용 네트워크인 SAN에 직접 연결된 공유 디스크를 활용하는 기법을 제시한다. 이 공유 디스크 공간은 파일 시스템 서비스 시작전에 전체 클러스터 내의 노드들이 공유할 수 있는 공간으로 특별히 설정해 둔 공간이며, 이 공간의 크기는 단순히 몇 개의 정수 값을 저장할 수 있는 작은 공간이다. 이 공유 디스크 공간은 SDB라는 이름으로 모든 노드들이 마운트하여 사용하게 된다.

SANique™의 회복 관리기 CRM은 다음과 같은 절차로 수행된다.

[오류 탐지 단계] 먼저 토큰 기반의 주기적 점검 메시지를 담당하는 오류 탐지 데몬 프로세스에 의해 초기에 구성된 클러스터 뷰와 현재 노드 뷰가 달라지면 회복 관리기 CRM에게 오류 탐지를 통보한다.

[그룹 형성 및 마스터 노드 선정 단계] 통신이 가능한 모든 노드들을 탐지하여 그룹을 형성하고, 이 중 한 노드를 선정하여 마스터 노드로 결정한다. 마스터 노드 결정방법은 투표(voting) 방법 등 여러 방법에 적용 될 수 있으나, 본 연구에서는 그룹 내의 노드들은 모두 동등 조건이므로, 노드 번호에 따른 순차적 선정 방법을 적용한다.

[그룹 정보 수집 단계] 모든 클러스터 노드들이 접근할 수 있는 SDB 공간을 활용하여 그룹들 간의 정보를 비교하기 위하여, 그룹 정보를 작성한다. 그룹 정보는 그룹 내의 노드 개수와 외부 네트워크와의 통신 여부로 구성된다. 외부 네트워크 통신 여부는 그 그룹이 서비스 그룹인지 오류 그룹인지를 판별하는 기준으로 사용한다. 외부 네트워크 통신 여부를 점검하기 위해서는 다양한 유틸리티들을 사용할 수 있다. 본 연구에서는 "ping" 명령어와 유사한 기능을 구현하여 게이트웨이로의 통신 점검 결과를 기반으로 통신 여부를 결정한다.

[그룹 정보 판독 및 기록 단계] 작성된 그룹 정보를 기록하기 위해 공유 디스크 보드 SDB를 접근한다. 이미 다른 그룹이 작성한 정보가 기록되어 있을 수 있으므로, 기록 전에 먼저 판독연산을 수행하여 다른 그룹의 정보를 읽어온다. SDB에서 판독한 다른 그룹의 정보가 자신이 작성한 그룹 정보 보다 좋은 상황이면, 자신의 그룹을 클러스터 내에서 제거(fence out)시켜야 하므로 공유 디스크 SDB의 기록

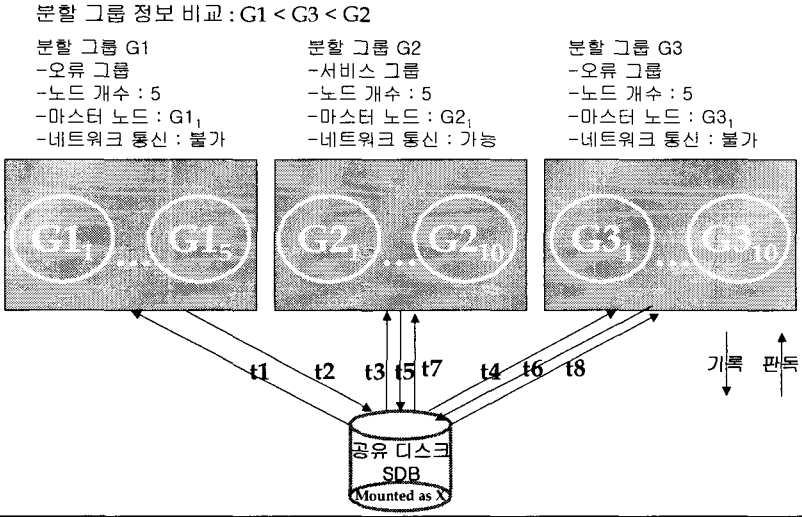
작업을 중단한다. 그렇지 않으면 적정 시간 후, 그룹 정보의 판독 및 기록 연산을 재수행한다. 두 그룹 간의 정보비교는 서비스 그룹이 오류 그룹보다 우세하며, 동등한 그룹 내에서는 노드의 개수가 많은 그룹이 우세한 것으로 판별된다.

[오류 처리 및 회복 단계] 재수행 후 자신의 그룹 정보가 공유 디스크 SDB상에 그대로 남아 있는 경우, 자신의 그룹이 분할그룹의 승자인 서비스 그룹으로 결정되고, 나머지 모든 그룹에 대한 오류 처리 작업을 수행한다. 다른 분할그룹들은 서비스 그룹의 정보를 판독하게 되어 모두 클러스터 내에서 제거된다.

4.2 그룹 정보 판독 및 기록의 직렬성

오류 탐지 및 회복 기법 중에서 그룹 정보 판독 및 기록 단계에서 또 다른 고려사항이 발생한다. 즉, 각 그룹들이 통신이 안되는 상태이므로, 공유 디스크 공간 SDB를 접근하는데 있어서 아무런 병행수행 제어가 적용되지 않아, 직렬성에 문제가 발생할 수 있다. 즉 판독연산과 기록연산을 수행하는 중간에 다른 기록 연산이 중복되어 기록될 수 있어, 그룹 정보 비교에서 잘못된 결과를 초래할 수 있다. SDB 내용을 판독하여 테스트하는 동안 다른 그룹이 자신의 정보를 기록하는 직렬성 위반 문제가 발생할 수 있다.

그림 8은 클러스터 노드들이 세 개의 분할그룹으로 나누어 졌을 때, 각 그룹의 마스터 노드들이 자신의 그룹 정보를 기록하고, 다른 그룹의 정보를 판독하여 서비스 그룹을 결정하는 예를 보이고 있다. 공유 디스크 SDB는 모든 노드에서 X라는 이름으로 마운트하여 사용한다. 먼저 시간 t1에서 분할그룹 G1이 공유 디스크 공간 SDB에 있는 기록을 판독($R_{G1}(X)$) 하지만, 초기값이므로 자신의 정보를 기록($W_{G1}(X)$) 한다. 그 후 시간 t3에 분할그룹 G2가 G1이 기록한 그룹 정보를 판독($R_{G2}(X)$)하고, 분할그룹 G2가 G1보다 좋은 조건임을 판단하게 된다. 그러나 G2가 자신의 그룹 정보를 기록하기 전에 시간 t4에 분할그룹 G3가 SDB의 정보를 판독($R_{G3}(X)$)하고, G3의 정보가 G1의 정보보다 우세하다고 판단한다. 시간 t5에 분할그룹 G2는 자신의 그룹 정보를 SDB에 기록($W_{G2}(X)$)하고, 이어서 시간 t6에 분할그룹 G3가 자신의 정보를 기록($W_{G3}(X)$)하게 된다. 같은 패턴으로 수차례 재수행 하더라도 분할그룹 G3는 G2보다 열악한 조건이어서 G2의 기록이 남아 있어야만 하지만, 판독-기록 순서상의 직렬성이 위반되어 G3의 그룹 정보가 공유 디스크 SDB에 남게 되고, G3는 자신의 기록이 남아 있음을 확인($R_{G3}(X)$)하고 결국 자신이 서비스 그룹인 것으로 판별하는 잘못된 결과를 유발하게 된다(시간 t8). 반면에 G2 입장에서 G3의 그룹 정보는 G2 자신의 그룹 정보보다 나쁘므로 자



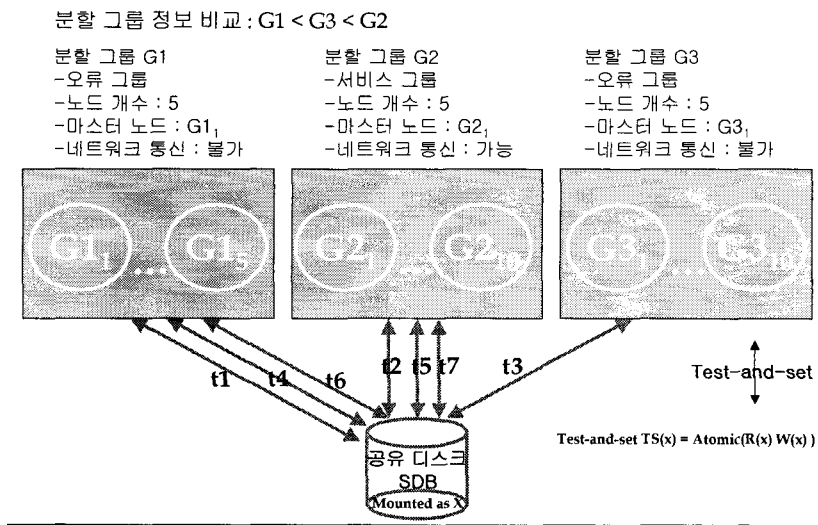
시간	t1	t2	t3	t4	t5	t6	t7	t8
연산	$R_{G1}(x)$	$W_{G1}(x)$	$R_{G2}(x)$ G1 보다 우세 판정	$R_{G3}(x)$ G1 보다 우세 판정	$W_{G2}(x)$	$W_{G3}(x)$	$R_{G2}(x)$ 오류 그룹으로 결정	$R_{G3}(x)$ 서비스그룹으로 결정

그림 8. 분할그룹 정보 비교의 직렬성 문제

신의 그룹 정보를 지속적으로 남기려고 하지만 최종적으로 자신의 그룹 정보가 아닌 G3의 그룹 정보가 남아있음을 확인(시간 t7의 $R_{G2}(X)$)하고 스스로 오류 그룹으로 판정하게 된다. 따라서, 분할그룹 G3가 G1, G2 그룹의 모든 노드들을 클러스터 내에서 제거하고 오류를 회복한 후, 파일 시스템 서비스를 재개하게 되면, 분할그룹 G2 보다 제한적인 파일 시스템 서비스를 하게 된다.

SANique™의 회복 관리기 CRM은 분할그룹 정보 비교시 판독후 기록의 연산 방법을 하나의 "test-and-set" 명령어를 사용한다. "test-and-set" TS(x) 연산은 R(x)W(x)이 원자적(atomic)으로 수행되는 것으로 정의된다. 따라서, 자신의 분할그룹 정보를 기록할 때, 기존 그룹정보를 비교 판단할 수 없고 단순히 가져오기 연산(fetch)만 가능하게 된다. 먼저 자신의 그룹 정보를 기록 한 후에 판독된 이 그룹정보는 자신의 그룹 정보와 비교에 사용된다. 즉, 판독-판단-기록의 연산 순서가 기록-판독-판단으로 변경되므로, 판단 후에 자신의 그룹 정보가 판독된 그룹 정보보다 우세하다고 판단되는 경우에만, 같은 작업을 재수행한다. 판독된 그룹 정보가 자신의 그룹 정보보다 우세한 경우에는 더 좋은 조건의 분할그룹이 존재한다는 것을 의미하므로 자신의 그룹에 해당하는 모든 노드에게 "fence out" 명령을 보내, 시스템내에서 스스로 제거된다. 재수행 후에 자신의 분할그룹보다 우세한 그룹이 없다고 판단되는 그룹은 서비스 그룹이 되고 나머지 그룹들은 클러스터에서 제거된다. 그림 9는 직렬성 문제를 해결하여 분할그룹 정보를 비교하는 과정을 보이고 있다.

그림 9의 분할그룹 G1이 시간 t1에 분할그룹 정보를 "test-and-set"을 통해 기록 및 판독하고, 시간 t2와 t3에 분할그룹 G2와 G3가 역시 기록-판독 연산을 수행한다. 판독된 정보는 각각의 마스터 노드에서 우열 비교를 판단하게 된다. 이 때, 분할그룹 G3는 G2의 정보를 판독하였으므로 패자로 결정되고, 그 그룹 내의 노드 $G3_1$부터 $G3_{10}$은 최종적인 서비스 그룹에 의해서 제거된다. 그러나 분할그룹 G1과 G2는 각각 그룹 정보 비교시 우세 결정을 하였으므로 시간 t4와 t5에 각각 기록-판단 연산을 재수행한다. 시간 t4에서 분할그룹 G1은 분할그룹 G3의 정보를 판독하였으므로, 자신의 그룹 정보가 우세하다고 판단하게 된다. 시간 t5에서 분할그룹 G2는 G1의 정보를 판독하여, 자신의 그룹이 우세하다고 역시 판단하게 된다. 그러나 시간 t6에서 분할그룹 G1은 G2의 그룹 정보를 판독하여 자신의 그룹을 패자 그룹으로 결정한다. 한편 시간 t7에서 분할그룹 G2는 최종적으로 자신의 그룹이 최종 서비스 그룹임을 결정하고 자신의 그룹 노드를 제외한 모든 노드들을 오류 처리하게 된다.



시간	t1	t2	t3	t4	t5	t6	t7
연산	$TS_{G1}(x)$	$TS_{G2}(x)$	$TS_{G3}(x)$ 오류 그룹으로 결정	$TS_{G1}(x)$	$TS_{G2}(x)$	$TS_{G1}(x)$ 오류 그룹으로 결정	$TS_{G2}(x)$ 서비스그룹으로 결정

그림 9. SANique™의 분할그룹 정보 비교 방법

따라서 제시된 방법을 통하여 분할그룹들 간에 최적의 그룹이 파일 시스템을 재개할 수 있으며, 또한 동시 접근 제어가 되지 않는 상태에서의 올바른 그룹 정보 비교가 가능하다.

4.3 SANique™의 회복 알고리즘

SANique™의 회복 관리기에서 제공하는 회복 알고리즘을 본 절에서 기술한다. 회복 알고리즘의 기본적인 전략은 앞 절에서 언급한 분할그룹 정보 비교를 통하여 서비스 그룹이 모든 나머지 분할그룹의 노드들에 대해 회복 절차를 수행하고, 패자 그룹은 클러스터 파일 시스템으로부터 제거되는 것이다. 본 회복 방법에 대한 의사 코드는 다음과 같다.

```
[1단계 : 오류 탐지]
1. 주기적 점검 메시지를 브로드캐스트 방식으로 전송한다.
2. if (초기 노드 뷰 != 현재 노드 뷰)
     2.1 CRM에 오류 상황을 통지하고, 현재 통신 가능한 노
         드 리스트를 전달한다.
     2.2 통신 가능한 노드들을 그룹화하고, 마스터 노드를 선
         정한다.

[2단계 : 분할그룹 문제 해결] (마스터 노드 수행 코드)
1. 자신의 그룹 정보(그룹 내 노드 개수 및 네트워크 상태 점검
   결과)를 작성한다.
2. 클러스터내의 공유 디스크인 SDB를 open 한다.
3. 공유 디스크 SDB에 작성된 그룹 정보를 기록-판독 한다.
4. for(i=0; i < 재수행 횟수; i++)
     4.1 if ( 자신의 그룹 정보 >= 판독된 그룹 정보)
                 sleep(1);
     4.2 else /* 패자 그룹 */
           4.2.1 그룹 내의 모든 노드들에게 I/O Fence Out
                 명령 전송
           4.2.2 클러스터 내에서 제거되기 위하여 I/O Fence
                 Out 루틴 호출
           4.2.3 시스템 패닉
5. if (판독한 그룹 정보 == 자신의 그룹 정보)
     서비스 그룹 = 자신의 그룹;

[3단계 : 회복 단계] (서비스 그룹 노드 수행 코드)
1. 서비스 그룹 노드를 제외한 모든 노드들의 기능을 전담하기
   위해 오류 노드들의 정보 수집
2. 오류 노드들의 기능을 서비스 그룹 노드들에 분담.
3. 현 서비스 그룹 노드들로 구성되는 클러스터 파일 시스템
   재개
```

위 알고리즘은 SANique™의 회복 관리기에 적용하여 실제 운영되고 있는 코드를 추상화하여 의사코드로 표현한 것이며, 수 십여개의 노드로 구성된 클러스터 내에서도 오류 노드의 탐지 및 회복을 수행할 수 있다. 본 제안된 방법은 분할 그룹으로 나뉘어진 노드들의 구성을 가장 최적화 하여, 최상의 상태의 그룹을 서비스 그룹으로 선정하는 방법으로서 기존의 임의 선정 방식에 의한 열악한 서비스 체제를 방지할 수 있다.

5. 결 론

본 논문에서는 SAN 기반의 클러스터 파일 시스템인 SANique™의 시스템 구성도 및 설계 내용을 기술하였으며, 특히 회복 관리기의 설계 내용에 대해 언급하였다. 클러스터 파일 시스템의 오류 상황 중에서도 분할그룹에 의해 발생하는 문제점을 조사하였고, 이를 해결하기 위하여 SAN을 통한 공유 디스크를 활용하는 방법에 대한 기법을 제시하였다. 제안한 기법은 시스템 서비스 온라인 상태에서 수행가능하며, 또한 오류 상황을 정확히 판단하기 힘든 분할그룹 상황에서도 최적의 그룹이 파일 시스템 서비스를 재개할 수 있는 회복 기법을 제시하였다. 본 논문에서 제안된 방법은 클러스터내의 노드 수가 많아질수록, 특히 분할그룹의 개수가 많아질수록 분할그룹 문제를 해결하는데 소요되는 시간이 길어지게 되며, 최적 그룹을 판단하는데 오차가 커지게 된다. 따라서 그리드 컴퓨팅과 같이 클러스터 노드 수가 많은 상황에서도 온라인 회복이 가능하도록 상수 시간 내에 오류 그룹을 탐지할 수 있는 방법을 계속 진행 중이다.

참 고 문 헌

[1] M. D. Dahlin, "Severless Network File Systems", Ph. D. Thesis at Computer Science Graduate Division of University of California at Berkely, 1999.

[2] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon, "Design and Implementation of the Sun Network File Systems", Proc. Of the Summer USENIX Conf. 1985.

[3] 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.

[4] VERITAS Software Corp., Veritas Volume Manager, http://www.veritas.com

[5] D. Teigland, "The Pool Driver : A volume Driver for SANs", Master's Degree Thesis, University of Minnesota, Dept. of Electrical and Computer Engineering, 1999.

[6] H. Maulshagen, "Logical Volume Manager for Linux", Sistina Technical Memo, http://www.sistina.com.

[7] MacroImpact, Inc., "SANique Cluster Volume Manager Functional Specification", MacroImpact Technical Memo, 2002.

[8] M. Satyanarayanan, "Scalable, Secure, and Highly Available Distributed File Access", IEEE Computer, 1999.

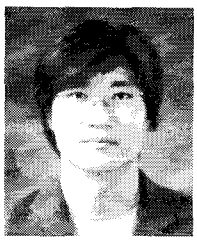
[9] K. W. Preslan, A. Barry, J. Brassow, R. Cattelan, A. Manthei, B. Marzinski, E. Nygaard, S. Oort, D. Teigland, M. Tilstra, S. Whitehouse and M. O'keefe, "Scalability and Failure Recovery in a Linux Cluster File System", Proc of the 4th Linux Showcase and Conference, 2000.

[10] U. Vahalia, Unix Internals : The New Frontiers, Prentice-Hall, NJ, 1999.

[11] C. C. Fan and J. Bruck, "The Raincore Distributed Session Service for Networking Elements", Proc. Of the International Parallel and Distributed Processing Symposium, 2000.

[12] P. S Weygant, "Primer on Clusters for High Availability", Technical Paper at Hewlett-Packard Labs, CA, 2000.

[13] P. T. murray, R. A. Fleming, P.D. Harry, P. A. Vickers, "Somersault : Enabling Fault-Tolerant Distributed Software Systems", TechnicalPaper HPL-98-81, Internet Comm. Systems Dept, Hewlett-Packard Labs. Bristol, 1998.



이 규 웅

1990년 한국외국어대학교 전자계산학과(이학사)
1992년 서강대학교 대학원 전자계산학과(공학석사)
1998년 서강대학교 대학원 전자계산학과(공학박사)
1998년~2000년 8월 한국전자통신연구원 인터넷서비스 연구부 선임연구원
2000년 9월~현재 상지대학교 컴퓨터정보공학부 조교수
관심분야 : 트랜잭션 처리, SAN 기반 자료저장 시스템, 분산 및 실시간 DB