

SAN기반 클러스터 파일 시스템의 일관성 검증을 위한 검사 프로그램의 설계 및 구현

백주현, 박성용
서강대학교 컴퓨터학과
rodent@dcclab.sogang.ac.kr, parksy@ccs.sogang.ac.kr
이장선, 오상규
(주)매크로임팩트
(sunny, sgoh)@macroimpact.com

Design and Implementation of Consistency Check Program for Storage Area Network Based Cluster File System

Joo-Hyun Baik, Sung-Yong Park
Dept. of Computer Science, Sogang University
Jang-Sun Lee, Sang-Gue Oh
Macroimpact, Inc.

요 약

본 논문에서는 기존 파일 시스템의 일관성 검사 프로그램 및 클러스터 파일 시스템의 특징 분석을 통해 클러스터 파일 시스템의 일관성을 검사하는데 필요한 프로그램 설계 요소 및 구현 방법에 대해 논한다. 나아가 이러한 설계 방법에 따라 구현된 프로그램을 실제 상용화된 클러스터 파일 시스템에 적용함으로써 검사 프로그램의 기능을 검증한다. 이를 위해 돌연변이 테스트(Mutation Test) 기법을 도입하여 일관성 검사 프로그램의 신뢰성 검증 방법을 고찰하고 그에 따른 테스트 결과를 살펴본다.

1. 서 론

비약적인 컴퓨터 성능의 발전에도 불구하고 디스크의 물리적 손상이나 정전과 같이 예기치 않은 문제가 발생할 경우 시스템이 유지하는 정보는 손상되어 질 수 있다. 이때 가장 큰 영향을 받는 것이 파일 시스템 정보이다. 따라서 시스템이 정상적으로 운영되는 것을 보장하기 위해서는 시스템 동작 전에 파일 시스템 내용에 손상된 부분이 있는지 검사하고 적절한 수리를 함으로써 디스크 정보의 일관성을 유지하는 기능이 필요하다.

멀티미디어 및 인터넷의 대중화를 통해 사용자들이 다루는 정보의 양이 급속히 증가함에 따라 노드간 통신 속도를 높이고, 운영체제의 버퍼링 등으로 인한 부하를 줄이기 위해 디스크와 같은 저장장치(Storage)를 노드간 시스템 네트워크에 직접 연결하는 방법으로서 SAN(Storage Area Network)[1]이나 NAS(Network Attached Storage)[2]가 등장하였다. 기존의 파일 시스템으로는 이러한 환경 변화에 대응하는데 한계가 있으므로 이에 적합한 파일 시스템에 관한 연구[3][4]가 시작되었고 그 대표적인 예가 클러스터 파일 시스템(Cluster File System)[5]이다. 클러스터 파일 시스템이 정상적인 서비스를 제공하기 위해서는 기존 파일 시스템에서와 같이 예기치 않은 파일 시스템의 손상에 대처하기 위한 방법이 요구된다. 따라서 기존의 파일 시스템 일관성 검사 프로그램들은 해당 파일 시스템 구조에 의존적이므로 클러스터 파일 시스템의 신뢰성을 검증할 수 있는 프로그램의 설계 방법에 관한 연구는 매우 중요하다고 할 수 있다.

본 논문에서는 리눅스 파일 시스템의 일관성 검사 프로그램

및 클러스터 파일 시스템의 특징 분석을 통해 클러스터 파일 시스템 일관성(Consistency) 프로그램의 설계 요소 및 구현 방법에 대해 논한다. 고려된 설계 요소들은 SANique[6]라는 상용 클러스터 파일 시스템의 일관성 검사 프로그램 구현에 적용한다. 또한 구현된 일관성 검사 프로그램의 검증을 위해 돌연변이 테스트(Mutation Test) 기법[7]을 도입하여 일관성 검사 프로그램의 신뢰성 검증 방법을 고찰하고 그에 따른 테스트 결과를 살펴본다.

본 논문은 다음과 같이 이루어져 있다. 2절에서는 리눅스 파일 시스템의 일관성 검사 프로그램인 fsck의 검사 단계를 분석하고 클러스터 파일 시스템의 구조를 살펴본 후 그로부터 클러스터 파일 시스템의 일관성 검사 프로그램이 포함해야 할 설계 요소를 고찰한다. 3절에서는 설계된 기능을 바탕으로 구현된 검사 프로그램을 실제 클러스터 파일 시스템에 적용하여 얻은 실험 결과를 보이고 구현된 검사 프로그램에 적용된 신뢰성 검증 방법을 설명한다. 4절에서는 결론을 맺는다.

2. SANique 클러스터 파일 시스템의 일관성 검사 프로그램

SANique 클러스터 파일 시스템[6]은 SAN 환경에서 여러 클라이언트가 동시에 파일을 접근할 수 있도록 해주고 디스크와 파일을 효율적으로 관리해 주는 시스템 소프트웨어로서 스트리밍, 이미징, 메일 서비스 또는 웹 서비스 등을 수행하는 파일 서버 클러스터에 최상의 성능과 가용성을 제공한다. 본 절에서는 기존 리눅스에서 사용된 파일 시스템 일관성 검사 프로그램의 절차를 알아보고 SANique 파일 시스템을 위한 일관성 검사에 사용된 방법을 기술한다.

2.1 리눅스 파일 시스템의 일관성 검사 단계

리눅스 파일 시스템 일관성 검사 프로그램(File System Consistency Check Program : fsck)은 리눅스 파일 시스템의 디스크 상 구조(On-Disk Structure)를 분석하여 손상된 부분이 있는지 검사하고 가능한 경우 수리를 하는 프로그램이다. fsck의 검사 단계는 다음과 같다.

첫 단계에서는 인덱스 노드 내의 메타 데이터 정보와 실제 데이터 블록 정보간의 일관성을 검사한다. 이 때 하나의 블록을 여러 개의 인덱스 노드가 가리키고 있는지에 대한 검사도 같이 이루어진다. 둘째 단계는 각 디렉토리 엔트리가 유지하는 내부 정보가 무결한지 검사한다. 셋째 단계는 디렉토리 연결 정보를 통해 트리(Tree)를 구성한 뒤 모든 엔트리들이 정상적으로 연결되어 있는지 검사한다. 넷째 단계에서는 각 파일과 디렉토리, 심볼릭 링크, 장치 제어 파일과 같은 특수 목적 파일들에 대한 참조횟수를 검사한다. 마지막으로 가용 영역에 대한 정보를 가지고 있는 비트맵 블록이 이전 단계들에서의 검사된 블록 사용 여부 정보와 일치하는지 검사하는 것으로 일관성 검사를 모두 마친다.

2.2 SANique 클러스터 파일 시스템의 구조

SANique 클러스터 파일 시스템은 <그림 1>에서와 같이 슈퍼 블록(Super Block), 비트맵 영역(Bitmap Area), 데이터 블록 영역(Data Block Area)으로 구성된다. 슈퍼 블록은 파일 시스템 전체를 관리하기 위한 정보를 저장한다. 비트맵 영역은 각 분산된 비트맵 영역에 접근하기 위한 비트맵 인덱스와 분산된 비트맵 영역으로 구성되며 각 비트맵에는 해당 분할 영역(Partitioned Area)상의 가용 블록 정보가 저장된다. 데이터 블록 영역은 사용자 데이터 블록과 파일이나 디렉토리 정보를 담고 있는 인덱스 노드 메타 데이터 블록으로 구성된다.

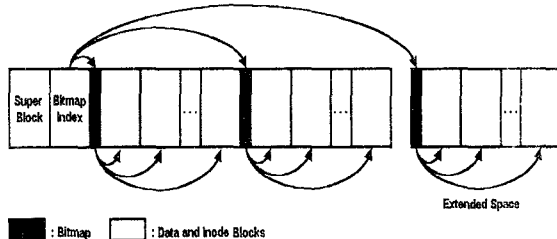


그림 1 SANique 클러스터 파일 시스템의 구조

기존의 파일 시스템에서는 비트맵 정보가 단일 파일 서버(Single File Server)의 메모리에 로드(Load)되어 관리되었다. 이는 여러 노드가 동시에 비트맵 정보에 접근하는 클러스터 환경에서 상당한 부하를 유발한다는 단점을 갖는다. 또, 리눅스 파일 시스템이 가지는 인덱스 노드와 데이터 블록간 연결 구조에서는 파일이 크기가 커질 경우 인덱스 노드와 데이터 블록간의 여러 단계의 계층 구조가 형성된다. 이는 데이터 블록들에 대한 일정한 접근시간을 보장하지 못하고 데이터 크기가 증가함에 따라 더 많은 접근 필요로 한다는 문제점을 갖는다.

위와 같은 문제점을 해결한 SANique 클러스터 파일 시스템의 일관성 검사 프로그램은 다음과 같이 설계되었다.

- 1) 기존 파일 시스템 검사 프로그램의 단위 기능 유지
- 2) 인덱스 노드 스템핑(Index Node Stuffing) 및 수정된 평형 구조(Modified Flat Pointing Structure) 확장 해싱(Extendible Hashing)에 의한 디렉토리 엔트리 관리 등 클러스터 파일 시스

템의 구조적 특성 반영

- 3) 직접적인 디스크 입출력 연산(Operation) 이용
- 4) 검사의 병렬성을 고려한 설계

2.3 일관성 검사 프로그램 단계

기존 파일 시스템들의 일관성 검사 단계에서는 인덱스 노드와 그것이 가리키는 데이터 블록들의 연결 관계에 대한 검사를 우선 수행하고 그로부터 파생된 정보들 통해 디렉토리 경로에 대한 검사를 수행하는 과정을 거친다. 클러스터 파일 시스템에는 기존의 파일 시스템에서와 같은 인덱스 노드 정보를 갖는 고정된 영역이 존재하지 않으므로 전통적인 일관성 검사 단계를 그대로 적용하기 어렵다. 따라서 각 인덱스 노드가 가지고 있는 데이터 블록들에 대한 검사를 위해서는 <그림 2>에서와 같이 디렉토리 구조에 대한 정보를 얻는 것이 선행되는 구조로 설계된다. 각 단계에서 얻어지는 정보는 컨텍스트 구조체(Context Structure)를 통해 해당 정보를 필요로 하는 단계에 전달된다.

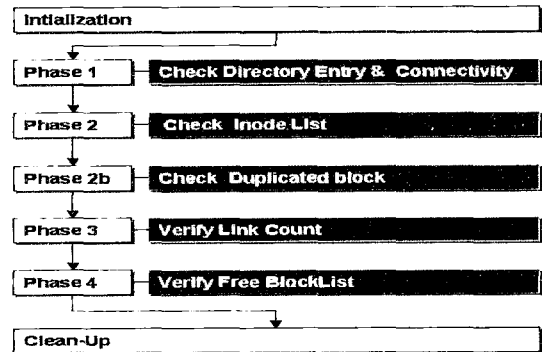


그림 2 클러스터 파일 시스템의 일관성 검사 단계

3. 구현 및 검증

3.1 실험 환경

설계된 일관성 검사 프로그램은 리눅스 커널 2.4.2 기반의 상용 클러스터 파일 시스템인 SANique 클러스터 파일 시스템 [6]을 대상으로 하여 C언어 통해 구현되었으며 각 노드간 네트워크는 하이버 채널을 사용했다.

3.2 뮤테이션 테스트를 이용한 검사 프로그램 검증

뮤테이션 테스트(Mutation Test)는 소프트웨어 테스트의 효과 분석을 위해 오래 전부터 사용되어온 기법이다. 원래의 뮤테이션 테스트는 테스트 케이스의 완전성 검증을 위해서 사용했지만 본 논문에서는 이를 응용하여, 일관성 검사 프로그램의 신뢰성을 검증하는데 이용한다. 뮤테이션 테스트 기법을 이용한 일관성 검사 프로그램의 테스트 방법은 <그림3>과 같다.

기존의 뮤테이션 테스트에서는 프로그램 자체에 뮤테이션 연산을 가해서 Mutant 프로그램을 만들어 내지만 본 논문에서는 데이터(파일 시스템 이미지)에 뮤테이션 연산을 가해서 뮤턴트 데이터(파일 시스템 이미지)를 생성한다. 만약 원본 파일 시스템과 뮤턴트 파일 시스템간의 차이를 일관성 검사 프로그램이 정확하게 지적해 낸다면 이 일관성 검사 프로그램은 뮤테이션 연산 O1, O2, ..., On에 대해서 완전하다고 말할 수 있다.

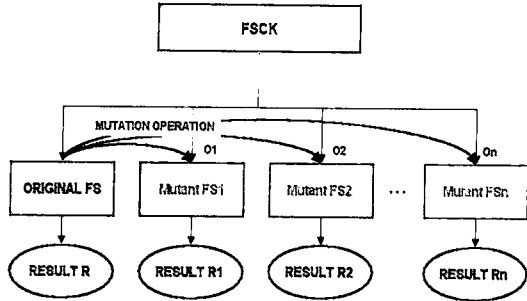


그림 3 일관성 검사 프로그램의 뮤테이션 테스트

3.3 실험결과

파일 시스템의 무결성을 보장하기 위해서는 다양한 뮤테이션 연산을 제공해야 한다. 본 논문에서는 기존 일관성 검사 프로그램이 제공하는 기능을 중심으로 <표 1>과 같은 뮤테이션 연산을 통해 테스트한다.

표 1 적용된 뮤테이션 연산 및 테스트 결과

	Mutation Operation	테스트 결과
1	Original 및 Duplicated Super Block 내 필드 정보 변경	Pass
2	Super Block 내 총 Extents 수 변경	Pass
3	Index Node의 mode 필드에 허용치 이상의 값 할당	Pass
4	Index Node의 size 필드와 할당된 Extent 수 필드값 변경	Pass
5	Index Node 내 link count 수 변경	Pass
6	Index Node에 연결된 데이터 블록 번호에 허용치 이상의 값 할당	Pass
7	Index Node에 의해 할당된 Bitmap 해제	Pass
8	두개 이상의 Index Node에 동일한 데이터 블록 번호 할당	Pass
9	..인 디렉토리 엔트리에 자신의 Index Node 번호 이외의 값 할당	Pass
10	..인 디렉토리 엔트리에 부모의 Index Node 번호 이외의 값 할당	Pass
11	디렉토리 엔트리의 ref_len 필드에 유효범위 이외의 값 할당	Pass
12	디렉토리 엔트리의 Index Node 번호 필드에 허용치 이상의 값 할당	Pass
13	디렉토리 엔트리의 name_len 필드에 유효범위 이외의 값 할당	Pass
14	디렉토리 엔트리의 Index Node 번호에 해당 하는 Bitmap 해제	Pass
15	생성된 Bitmap 해제	Pass
16	디렉토리인 Index Node의 데이터 블록 번호 삭제	Pass

실험은 각 뮤테이션 연산을 100회씩 반복하여 구현된 일관성 검사 프로그램이 해당 뮤테이션 연산에 의한 파일 시스템의 인위적 손상을 몇 번 검출해 내는가를 알아보는 형태로 이뤄졌다. 매 번의 검사에 대하여 뮤테이션 연산이 변형시킨 파일 시스템의 내용이나 위치는 해당 뮤테이션 연산의 제한 조건을 만족하는 임의의 값을 취한다. <표 1>에서 볼 수 있듯이 구현된 일관성 검증 프로그램은 뮤테이션 테스트에 대하여 신뢰성 있는 결과를 나타낸다.

4. 결론

시스템 정보의 손상이 발생하기 쉬운 상황 하에서도 운영체제가 신뢰성 있는 서비스를 제공하기 위해서는 파일 시스템의 일관성 검사가 필수적이다. SAN 기반 클러스터 파일 시스템이 향상된 노드 연결 기술 및 저장 기술을 통해 빠른 서비스를 보장하기 위해서는 기존 파일 시스템에서와 같이 예기치 않은 손상에 대한 대처 방법이 요구된다.

본 논문에서는 기존 파일 시스템 검사 방법의 분석 및 클러스터 파일 시스템의 특징 고찰을 통해 클러스터 파일 시스템을 위한 일관성 검사 프로그램을 설계하고 구현함으로써 SAN 기반 클러스터 파일 시스템이 신뢰성 있는 동작을 할 수 있는 방안을 제시하였다. 또한 구현된 검사 프로그램의 검증 방법으로 서뮬레이션 테스트를 도입하였다. 이는 향후 다양한 클러스터링 환경 하에서 동작하는 파일 시스템의 일관성 검증에 있어서 효과적인 설계 기준 및 테스트 방안으로 이용될 수 있다. 실험 결과를 통해 구현된 일관성 검사 프로그램이 다양한 뮤테이션 연산에 대해 신뢰성 있는 검출 능력을 보임을 알 수 있었다.

클러스터 파일 시스템은 각 노드가 병렬적으로 동작할 수 있는 환경에서 운영된다. 본 논문이 설계 단계에서 언급한 것과 같이 일관성 검사 작업을 각 노드에 분배함으로써 얻어지는 검사 성능의 향상에 관한 추가적인 연구가 요구된다. 클러스터 파일 시스템의 일관성 검사나 그에 대한 테스트 기법은 아직 초기 연구 단계이나 향후 더욱 안정적인 클러스터링 환경을 구축하기 위해서는 본 논문에서 도입한 테스트 연산 외에도 다양한 테스트 기법에 대한 연구가 필요하다.

참고문헌

- [1] Randy H. Katz, "High- Performance Network and Channel Based Storage", Proceedings of IEEE, Vol.80, No.8, pp.1238- 1261, 1992.
- [2] G. A. Gibson, R. V. Meter, "Network-Attached Storage Architecture", Comm. of the ACM, Vol. 43, No 11, November, 2000.
- [3] T. E. Anderson, M. Dahlin, et al., "Serverless Network File Systems", ACM Transactions on Computer Systems, pp. 41-79, 1996.
- [4] C. A. Thekkath, T. Mann, and E. K. Lee, "Frangipani: A Scalable Distributed File System", Proceedings of the 16th ACM Symposium on Operating System Principles (SOSP), pp.224-237, 1997.
- [5] Kenneth W. Preslan, Andrew Barry, Jonathan Brassow, et al., "Scalability and Failure Recovery in a Linux Cluster File System", Proceedings of the 4th Annual Linux Showcase and Conference, Atlanta, October 10-14, 2000, Atlanta, Georgia, USA
- [6] <http://www.macroimpact.com>
- [7] A.Jefferson Offutt, "A Practical System for Mutation Testing ; Help for the Common Programmer", Twelfth International Conference on Testing Computer Software, pages 99--109, Washington, DC, June 1995.