

SANtopia를 위한 성능 감시 및 구성 관리 도구의 설계 및 구현[☆]

Design and Implementation of a Performance Monitoring and Configuration Management Tool for SANtopia

임기욱* 나용희** 민병준*** 서대화**** 신범주*****
Kee-Wook Rim Yong-Hi Na Byoung-Joon Min Dea-Wha Seo Bum-Joo Shin

요약

상대적으로 느린 저장 장치들의 입출력 처리 속도는 컴퓨터 시스템의 전체적인 성능을 저하시키는 요인이 된다. 이를 개선하기 위한 한 방안으로 SAN(Storage Area Network)이 제안되었다. 파이버채널과 같은 고속 네트워크에 저장 장치가 직접 연결된 SAN 시스템을 효율적으로 이용하기 위해서는 적절한 관리 도구가 필요하다. 본 논문에서는 SAN 환경에서 전역 파일 공유를 제공하는 SANtopia 시스템을 위한 성능 감시 및 구성 관리 도구의 설계 및 구현 결과에 대하여 논한다. 개발된 도구는 SANtopia 시스템을 구성하는 리눅스 호스트의 성능을 감시하고 호스트 및 저장 장치들의 구성을 관리하기 위한 것이며 자바 언어를 이용하여 GUI(Graphic User Interface)를 제공한다.

Abstract

I/O processing speed of relatively sluggish storage devices incurs overall performance degradation in computer systems. As a solution to improve the situation, SAN(Storage Area Network) has been proposed. In order to utilize a SAN system more effectively, where storage devices are directly connected with a high speed network such as Fibre Channel, a proper management tool is needed. In this paper, we present a design and implementation of a performance monitoring and configuration management tool for the SANtopia system which supports global file sharing in the SAN environment. The developed tool is to monitor the performance of the Linux hosts composing the SANtopia system and to manage the configuration of the hosts and storage devices. It also supports GUI(Graphic User Interface) environment using the JAVA programming language.

· 키워드 : SAN(Storage Area Network), Network Distributed Systems Management, Linux Host, Java Language

1. 서론

고성능 네트워크와 대규모 저장 장치의 발

달은 여러 대의 단일 컴퓨터의 집합체 즉, 클러스터링 환경을 이끌어냈다. 클러스터링은 여러 대의 독립 컴퓨팅 노드들을 고속 및 고성능 네트워크로 묶어 마치 하나의 컴퓨팅 노

☆ 본 연구는 한국전자통신연구원 위탁연구, 인천대학교 RRC, 선문대학교 교내연구 지원에 의해 수행되었음.

* 비회원 : 선문대학교 지식정보산업공학과 교수
rim@sunmoon.ac.kr (제 1 저자)

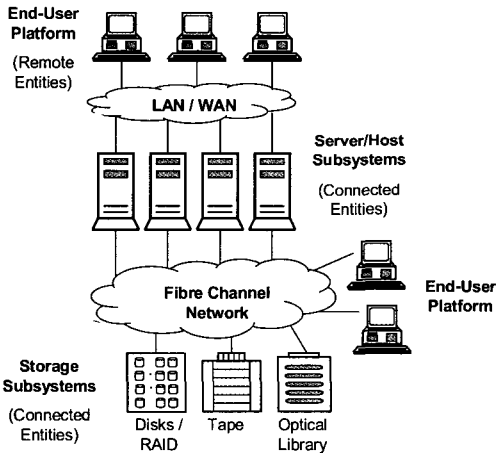
** 비회원 : 인천대학교 컴퓨터공학과 (석사과정)
ramen@incheon.ac.kr (공동저자)

*** 정회원 : 인천대학교 컴퓨터공학과 부교수
bjmin@incheon.ac.kr (공동저자)

**** 비회원 : 경북대학교 전자전기공학부 교수
dwseo@ee.knu.ac.kr (공동저자)

***** 비회원 : 밀양대학교 컴퓨터공학과 교수
bjshin@mnu.ac.kr (공동저자)

드처럼 취급하는 것으로 기존의 단일 컴퓨터로는 불가능하던 고도의 컴퓨팅 능력을 이끌어 내는 것이다. 이미 알려진 바와 같이 컴퓨터 저장 장치의 처리속도는 CPU나 메모리와 같은 다른 컴퓨터 부품의 발전 속도를 따라가지 못하고 있다. 고성능 컴퓨팅 환경에서의 이러한 저장 장치의 입출력 속도에 따른 병목 현상을 해결하기 위해 여러 가지 방법들이 시도되어 왔다. 그 가운데 대용량 저장 장치들을 파이버 채널로 묶어 저장 장치의 고성능화를 꾀하는 SAN(Storage Area Network) 환경이 제안되었다. SAN은 대용량 저장 장치들을 파이버 채널이라는 고속 네트워크 기술을 이용해 서로 묶은 것이다. 그림 1은 전형적인 SAN 환경을 나타낸 것이다[1,2,3,4].



(그림 1) SAN(Storage Area Network) 환경

SAN은 파이버 채널(Fibre Channel)로 디스크, 테이프, 광 라이브러리와 같은 저장 서버 시스템과 이들을 여러 사용자들이 공유할 수 있도록 해주는 서버 또는 호스트를 직접 연결한다. 여기에 관리자 또는 일반 사용자를 위한 단말이 연결된다. 원격 사용자는 기존의 LAN이나 WAN을 통하여 호스트의 도움을 받아 저

장 장치에 접근할 수 있다. 이는 서버에 디스크를 직접 연결한 기존의 DAS(Direct Attached Storage)와 달리 저장 장치를 고속의 파이버 채널에 직접 연결시켜 사용할 수 있기 때문에 성능, 확장성(scalability), 가용성 면에서 매우 우수하다.

이러한 SAN 환경에서 효율적인 시스템의 이용을 위해서는 적절한 관리 도구의 사용이 필수적이다. 점점 성능이 발달하고 여러 가지 기능들이 추가됨에 따라 전체적인 시스템의 성능 자체의 개선이 있었지만, 개선된 성능들을 최대로 이용하기 위한 관리의 측면에 있어서는 여전히 부족하고 어려운 점이 많다 [5,6]. 구현된 시스템 자원의 효율적 이용과 관리를 단순 명령 창 터미널이나, 개별적인 환경 설정 파일 등을 관리자가 손수 설정하는 등의 방식으로로는 한계가 있고, 그 노력에 비해 효율성이 떨어진다. 게다가 표준화 작업의 미흡으로, 특정 표준에 기반을 둔 관리 도구의 개발과 사용보다는 기존 스토리지 시스템 관리 소프트웨어를 개발, 보급하고 있는 업체들을 중심으로 개발되고 상용화되어있는 실정이다. 개발된 관리 도구들 또한 자사 제품에 특화되어 있어 여전히 범용으로 사용하기에는 문제점이 많다. 또한 그러한 제품들은 대부분 시스템 의존적인 프로그래밍 언어로 작성되었기에, 다른 환경으로의 이식과 사용이 용이하지 않다.

이에 본 논문에서는 SAN 환경의 감시 및 관리 작업을 보다 용이하고 효율적으로 수행할 수 있는 새로운 SAN 시스템 관리 도구의 개발에 대해서 논한다. 이는 한국전자통신연구원(ETRI)에서 개발 완료 단계에 있는 SAN 시스템인 SANtopia 연구 개발의 일환으로 진행된 것이다.

본 논문의 2장에서는 지금까지 SAN 관리를 위한 노력으로 발표된 관리 소프트웨어의 현황을 정리한다. 본 논문에서 제시하는 관리 도

구의 역할의 이해를 돕기 위하여 3장에서는 현재 개발 완성 단계에 와 있는 SANtopia 시스템을 간략히 소개한다. 4장에서는 SANtopia 시스템의 성능을 감시하고 구성을 관리하기 위한 도구의 설계 내용을 설명한다. 5장에서 이 관리 도구의 구현 결과를 보이고, 마지막으로 6장에서 본 논문의 결론을 맺는다.

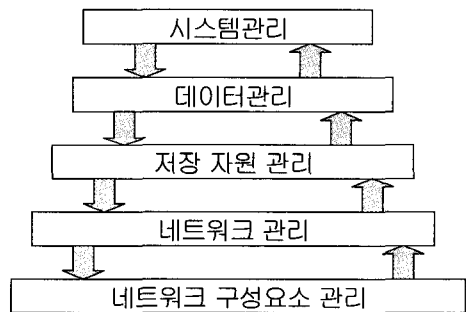
2. SAN 관리를 위한 기존 연구 및 산업계 동향 분석

지금까지 파이버 채널을 기반으로 한 SAN 환경에 있어서의 표준화 작업은 서버 시스템과 저장 장치에서의 파이버 채널 인터페이스 사양에 대하여는 어느 정도 업계 표준이 자리 잡고 있는 실정이지만 SAN 환경을 관리하는 기본 기능이나 이를 구현하는 관리 소프트웨어에 대한 표준화 작업은 아직 미비한 실정이다.

SAN 환경에 대한 관리 소프트웨어의 구현은 기존의 저장 장치 시스템 관리 소프트웨어를 개발 보급하고 있는 업체들을 중심으로 개발되어 상용화되고 있다. 그 종류와 기능이 다양하지만, 대표적인 SAN 관리 소프트웨어로 Tivoli사의 Vision (Storage management solutions)[7], Sun사의 StoreEdge[8]와 Compaq사의 SANworks Storage Management Software[9], 그리고 HP사의 HP SureStore E와 SAN Manager LM, SAN Manager DM과 Legato사의 ESMA와 Networker, GEMS, SmartMedia[10], Vixel사의 Insite2000[11], Veritas사의 SANPoint Control, HighGround Systems사의 SRM(Storage Resource Management) [12], Gadzoox사의 Ventata SAN manager[13], Brocade사의 WEB TOOLS[14], Datacore사의 SAN Cantata software와 QLogic사의 QLconfig와 QLview 등을 들 수 있다. 이들 중 Tivoli사와 같은 업체는 IBM사와 함께 SNIA(Storage Networking Industry Association)나 FCIA

(Fibre Channel Industry Association)같은 SAN 기술과 관련된 표준화 기구에 적극적으로 참여하고 있다.

SAN 환경에 대한 관리 소프트웨어들은 데이터 백업이나, 저장 상태 모니터링, SAN 구성요소들 간의 연결상태나 장치들을 나타내는 구성도의 제공 등 SAN 관리에 있어서 필수적인 기능들을 대부분 포함하고 있다. 그러나 각 개발업체의 동향이나 개발의도에 따라서 독특한 관리기능들을 포함하고 있기도 하다. SAN 환경 관리를 위한 프레임워크를 그림 2와 같이 다섯 개의 계층으로 나눌 수 있다.



(그림 2) SAN 관리 프레임워크

네트워크 구성요소 관리에서 고려되어야 하는 관리기능들은 적절한 성능유지를 위해 작업처리율이나 데이터 전송률, I/O 동작속도를 측정해서 정보를 수집하고 분석하는 기능과, 결함 발생을 차단하기 위해 어댑터와 그에 연결된 장치들의 연결상태를 체크하고 진단하는 기능과 고장횟수를 카운트하는 기능이다. 또한 구성관리 측면에서는 물리적인 SAN 환경의 위치 구성도를 제공하고, 장치 가격이나 관리 경향 같은 자산 정보를 수집하는 기능이 요구된다.

네트워크 관리에서 고려되어야 할 관리 기능들은 성능관리를 위해 네트워크의 평균 처리율 정보를 수집하고 파일 크기 같은 성능에 영향을 미치는 요인들을 분석해보며 트래픽의 패턴도 분석한다.

또한 결함의 허용을 위해서는 네트워크의 고장을 보고하는 기능과 연결 상태가 불안정하거나 고장이 난 경우에는 이벤트를 통보하는 기능, 고장 횟수를 카운트하는 기능 등이 필요하고 SAN의 구성을 관리하기 위해서는 네트워크의 구성상태, 즉 구성도를 제공하는 기능이 요구된다.

저장 자원 관리에서 고려되어야 할 관리 기능은 데이터와 저장 장치 등의 적절한 성능 유지를 위해서 전체적인 데이터 증가율과 저장 용량 증가율을 측정하는 기능, 저장 장치 사용패턴을 분석하는 기능, 장치와 경로간의 부하를 분산하는 기능 등이 요구된다. 또한 결함 관리를 위해 저장 시 병목현상과 스트레스 지점을 탐색하고 재난 등에 대한 대비책으로 저장을 복제하는 기능이 필요하며 구성관리 면에서는 디스크 같은 저장 장치의 파티션 구성도를 제공하고 자동적으로 디스크 할당과 교체를 수행하는 기능 호스트와 LUN 사이의 논리적 연결을 유지하는 기능과 디스크 중 사용하지 않는 공간을 탐지하는 기능 등이 요구된다.

데이터 관리에서 고려되어야 할 관리기능은 성능 관리를 위해 데이터 처리율을 측정하는 기능이 필요하고 주기적인 백업과, 동기 혹은 비동기 방식으로 데이터를 복제하는 기능, 재난 복구에 대한 계획을 수립하는 기능 등 결함관리를 위한 기능이 필요하다. 또한 구성관리를 위한 기능은 저장 용량의 계층적 구조를 유지하는 기능과, 데이터에 대한 소유자, 허가권, 잠금에 대한 정보를 유지하는 등의 기능이 요구된다. 데이터 관리 측면에서 중요시 되는 관리기능은 여러 가지 백업 방법을 이용해서 데이터를 보호하고, 원격지 백업을 위한 데이터 미러링과 재난 복구에 대비하기 위한 데이터 복제를 하는 기능이 필요하다. 그 밖의 성능관리나 구성관리는 데이터 관리 계층에서는 많이 고려되지 않고 있는 실정이다.

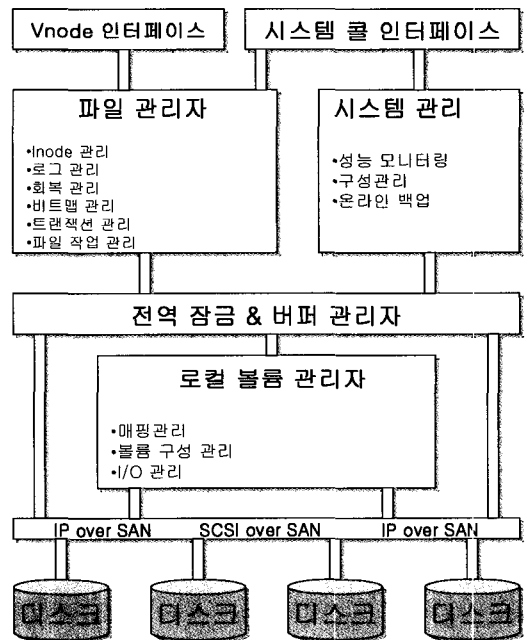
시스템 관리 측면에서 요구되는 관리 기능들은 클라이언트와 서버의 처리율이나 가장 빠른 클라이언트, 가장 느린 클라이언트를 측정하는 성능 관

리기능 등이 필요하다. 그러나 시스템 관리 측면은 대부분의 SAN 관리 소프트웨어에 포함되지 않고 있다.

본 논문은 위에서 열거한 기능 중 저장 자원의 구성 관리 영역과 시스템의 전체적인 성능 관리 영역에 초점을 두고 있다. 기존의 관리 소프트웨어와는 다르게 본 도구는 자바 언어를 기반으로 함과 동시에 관리 대상이 되는 SAN 시스템과의 인터페이스를 파일 접근이나 유틸리티 구동의 수준에서 이루어지게 하여 이식성이 매우 우수하다.

3. SANtopia 시스템의 개요

SANtopia 시스템은 리눅스용 네트워크 연결형 자료저장 시스템 소프트웨어로 SAN 환경에서 대용량 파일에 대한 전역 파일 공유 기능을 지원하기 위해 한국전자통신연구원에서 개발해 온 것이다 [15].



(그림 3) SANtopia 구조

그림 3에 나타낸 SANTopia는 각 기능에 따라 볼륨 관리자, 버퍼 관리자, 잠금 관리자, 파일 관리자로 구성된다.

■ 볼륨 관리자

볼륨 관리자는 SAN에 연결된 모든 디스크를 모아서 하나의 관리 대상으로 볼 수 있도록 하고 사용자가 원하는 방식에 따라 이것을 분할하여 볼륨을 구성하고 상위레벨의 모듈에서 이 볼륨 위에 I/O를 실행할 수 있도록 해주는 모듈이다.

■ 버퍼 관리자

기존 리눅스 시스템들은 각 호스트들이 독립적으로 운용되는데 비해서 SANTopia에서는 다수의 리눅스 호스트가 클러스터 형태로 운용된다. 클러스터 환경에서 시스템의 성능향상을 위해서는 버퍼의 통합관리가 필수적이다. 즉, 개별 호스트에 있는 버퍼들을 통합해서 전역 버퍼로 관리함으로써 버퍼의 자료 적중률을 높여야 한다. 버퍼 관리자는 각 호스트에 산재해 있는 버퍼에 어떤 내용이 배치되어 있는지 관리하고 이를 바탕으로 호스트 간에 버퍼의 내용을 주고받음으로써 전체 클러스터의 디스크 접근 횟수를 줄이고 성능을 향상시키는 역할을 담당한다.

■ 잠금 관리자

SANTopia에서는 다수의 호스트가 동시에 자료를 사용하기 때문에 자료의 일관성을 위해서 잠금 기능이 필요하다. 잠금 관리자는 호스트 간의 자료에 대한 경쟁이 발생할 때 접근 순서를 결정해주는 역할을 담당함으로써 자료의 일관성을 보장함은 물론 데드락 및 기아(starvation) 상황이 발생하지 않도록 하는 기능을 담당한다.

■ 파일 관리자

리눅스에서는 새롭게 개발된 파일 시스템을 지원하기 위한 방법으로 가상 파일 시스템을 사용한다. 파일 시스템의 특성에 맞게 각 파일에 대한 연산

을 구현하고 이를 가상 파일 시스템 인터페이스와 연결하여 사용자들이 파일 시스템의 종류에 영향을 받지 않고 자유롭게 파일에 대한 연산을 수행할 수 있다.

본 논문에서 제안하는 SANTopia 성능 감시 및 구성 관리를 위한 도구는 전체 SANTopia 시스템을 감시하고 관리하기 위해 리눅스 호스트와 관리자용 컴퓨터 노드에 설치되어 동작한다. 관리 도구는 기본적으로 서버 클라이언트 구조로 이루어져 있으며 이들 사이의 통신은 기본적으로 TCP 프로토콜을 이용해 이루어진다.

4. SANTopia 관리 도구의 설계 및 구현 방법

이 장에서는 SANTopia 시스템 성능 감시와 구성 관리를 위한 관리 도구의 설계와 구현 방안을 설명한다. 우선 설계 목표를 제시하고 그에 따른 전반적인 관리 시스템의 구조, 그리고 SANTopia와 관리 시스템과의 인터페이스를 설명한다. 마지막으로 관리 시스템을 구성하는 소프트웨어 모듈 구현을 위한 클래스 구성에 대하여 논한다.

4.1 설계 목표

본 논문에서 결과로 제시한 연구의 목적은 SAN에 연결된 저장 장치들의 효율적인 공유와 이용 및 관리를 위한 전체 시스템의 성능 감시 및 구성 관리의 기능을 설계하고 이에 대한 프로토타입을 개발하는 것이다. 여기에 기존의 리눅스 환경이나 SAN 시스템 관리도구들이 가지는 문제점들, 명령어 타이핑 방식이나 직접 설정 파일의 수정을 통한 시스템 관리 방식의 불편함, 특정 시스템에 의존적인 환경으로 인한 타 시스템으로의 이식 및 사용의 어려움, 시스템마다 다른 관리 환경으로 인한 사용자의 작업 효율 저하 등을 개선하기 위해

시스템 독립적인 개발언어인 자바를 사용해 향후 관리 도구의 이식의 용이함과 최종 사용자에게 동일한 사용 환경을 제공함으로써 작업의 능률성 향상 제공을 목적으로 한다. 특히 텍스트 방식이 아닌 GUI(Graphic User Interface) 방식의 환경을 제공함으로써 관리 및 이용에 좀 더 편리함을 제공하는 것이다.

본 관리 도구의 주요 기능은 다음과 같다.

■ 호스트 성능 감시

- 호스트의 CPU, 메모리 사용량, 프로세서, 디스크, 네트워크 이용 상태 감시

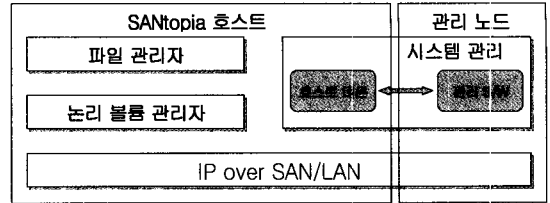
■ 구성 관리

- 호스트 매핑
- 논리적 공간 생성/등록/삭제/제거
- 파일 시스템 생성/등록/삭제/제거

4.2 관리 구조

SANtopia 시스템 관리를 위한 전체적인 구조는 그림 4에 나타낸 바와 같다. 시스템 관리자는 크게 두 개의 모듈로 나누어진다. 하나는 사용자(또는 관리자)와의 인터페이스를 통해서 관리 정보를 그래프나 테이블 형태로 나타내고 호스트 제어 명령을 전달하는 관리 소프트웨어이고 또 다른 하나는 관리 소프트웨어의 에이전트 역할을 하는 호스트 데몬이다. 호스트 데몬은 SANtopia 시스템을 구성하는 여러대의 리눅스 호스트들에 탑재되어 파일 관리자, 논리 볼륨 관리자, 그리고 리눅스 운영 체제를 통해서 필요한 정보를 수집하고 여과한 후 관리 소프트웨어에 전송한다. 또한 사용자의 명령에 따라 관련 파라미터 값들을 설정할 수 있다. 그림에서 관리자 소프트웨어는 별도의 관리자 노드에 탑재되어 SAN 또는 일반 LAN으로 연결된 것으로 표현되어 있으나 반드시 별도의 물리적 노드

로 분리되어야 하는 것은 아니다.



(그림 4) SANtopia 시스템 관리 구조

4.3 SANtopia 호스트와의 인터페이스

시스템 관리자의 호스트 측 소프트웨어 모듈인 호스트 데몬이 SANtopia와 인터페이스 하는 부분은 둘로 나누어 생각할 수 있다. 하나는 시스템의 성능을 감시하기 위해 호스트의 성능 관련 정보를 수집하는 것이고, 다른 하나는 구성 관리를 위해 관리자의 명령에 따라 하부 구조의 해당 유틸리티를 호출하는 것이다. 여기서 하부 구조란 호스트 내의 파일 관리자나 논리 볼륨 관리자가 제공하는 인터페이스를 의미한다.

■ 리눅스 호스트 정보 수집

SANtopia 시스템을 모니터링 하기 위해서는 적절한 시스템 정보를 얻어오는 수단이 필요하다. 과거에는 시스템 정보를 구하기 위해 ps 나 uptime 같은 프로그램이 커널 데이터 구조에 직접 접근했으나, 이는 커널 외부에 대한 지식을 요구했고 보안 침범에 대한 우려도 있었다. 하지만 최근의 시스템은 시스템 상태 정보를 얻을 수 있는 /proc 파일 체계를 가지고 있다. 이런 파일은 일반적으로 텍스트 파일이며, 리눅스는 이 /proc 파일 체계를 통해 이전보다 더 많은 시스템 정보를 제공하고 있다. /proc 디렉토리 아래에는 특정 시간에 시스템에서 실행중인 각 사용자 단계 프로세스를 위한 디렉토리가 있다. 이 디렉토리의 이름은 프로세스 번호를 10진수로 표현한 것이다. 또한, /proc/self

는 현재 프로세스의 디렉토리에 대한 기호 링크이다. 이 링크는 각 프로세스마다 다르다. 이런 디렉토리 내에는 다수의 파일이 있는데, 여기에서 나오는 \$pid는 원하는 프로세스의 프로세스 id로 대체하면 된다.

■ SANtopia 구성 관리를 위한 유틸리티

SANtopia의 파일관리자와 논리볼륨관리자는 시스템 관리자가 사용할 수 있는 다양한 유틸리티를 제공한다. 저장 풀을 구성하는 모든 호스트에 대한 매핑 구성 정보, 논리 볼륨의 구성 정보, 파일 구성 정보 등은 각각의 파일로 저장된다. 다음은 SANtopia 볼륨을 관리하기 위한 유틸리티들의 예이다.

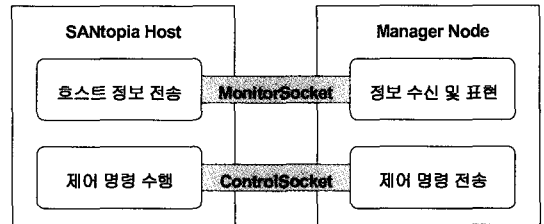
- create : SANtopia 볼륨을 생성
- load : 생성된 볼륨을 호스트 커널에 등록
- unload : 커널에서 볼륨 등록정보 제거
- destroy : 해당 볼륨의 label 정보를 삭제, 볼륨을 공유하는 호스트 중 하나라도 loading하여 사용하고 있다면 label 정보를 삭제할 수 없음
- grow : 물리 볼륨의 온라인 확장, 새로운 디스크 파티션을 추가하고, 구성 정보 변경을 label에 반영
- shrink : 물리 볼륨의 온라인 축소, 사용되지 않는 장치를 제거하고, 볼륨 구성 정보의 변경을 반영
- snapshot : 스냅샷 생성 및 삭제
- volinfo : 현재 호스트에 등록된 모든볼륨의 구성 정보 제공

관리 도구에서는 GUI를 통해 정보를 제공하며 관리자의 구성 관리를 위한 제어 명령 수행을 위하여 위와 같은 유틸리티 중 해당 내용을 호출한다. 관리자는 그 관리 수행 결과를 변경된 시스템 정보가 화면에 나타남으로써 확인할 수 있다.

4.4 클래스 구성 및 구현 방안

■ 호스트와 관리자 노드 간의 통신

앞서 그림 4에 나타낸 바와 같이 시스템 관리자를 위한 관리 도구의 전체적인 구조는 호스트 데몬과 구성 관리 소프트웨어 모듈로 이루어진다. 이들 사이의 통신을 위해 TCP/IP 프로토콜을 사용한다. 이를 처리해 주기 위한 소켓 처리부가 프로그램 내부에 구현되어 있다. 그 전체 구조는 다음 그림 5와 같다.



(그림 5) 관리를 위한 통신 구조

우선, 호스트에서 데몬 프로그램을 구동시킨 후, 관리자 노드에서 프로그램을 구동하면 MonitorSocket을 통해 호스트 정보가 관리자 노드로 전달되어 사용자 작업 창에 출력되도록 한다. 사용자가 호스트에 대하여 볼륨 생성이나 삭제, 파일 시스템 생성 등과 같은 특정한 작업을 하려고 할 때 사용자 작업 창에서 설정 명령을 보내면, 이들 사이의 ControlSocket을 통해 호스트로 제어 명령이 전달되어 호스트 상에서 그 명령이 수행된다. 이처럼 기본적으로 소켓을 이용한 명령 또는 자료의 송수신 구조를 취한다. 작업 처리 요청 및 자료 송수신을 위한 통신은 TCP 프로토콜을 이용해 구현되었다.

■ 호스트 데몬 모듈

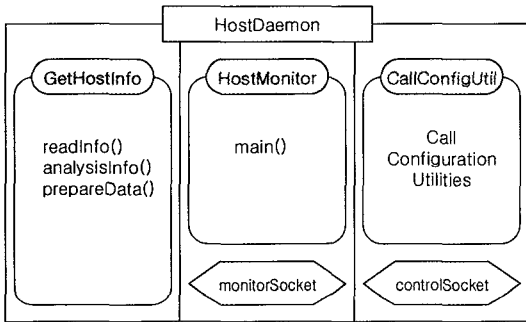
호스트 데몬 모듈은 SANtopia 시스템의 각 호스트에서 구동되며 프로그램 수행 후 데몬 형식으로 수행되어 관리자 노드로 시스템 관련 정보를 전송하고 관리자 쪽에서 SANtopia 시스템 관리 요청

이 들어오면 그 작업을 처리해주게 된다.

호스트 데몬의 임무는 다음과 같다.

- 호스트 시스템 정보 획득
- 관리자 소프트웨어 모듈과의 송수신
- 제어 명령에 따른 별도의 관리 유틸리티를 호출

호스트 데몬 프로그램은 그림 6에 나타낸 바와 같이 기본적으로 HostMonitor, GetHostInfo, CallConfigUtil의 세 클래스로 구성되어 있다.



(그림 6) 호스트 데몬 모듈의 클래스 구성

HostMonitor 클래스는 호스트 노드에 상주하면서 관리자 노드의 HostView가 가동되면 일정 시간마다 GetHostInfo를 실행하여 호스트 정보를 읽어 monitorSocket을 통해 전송한다. HostMonitor 클래스는 호스트 전체적인 프로그램을 구동시키는 부분으로서, 자바 main() 메소드를 포함하고 있다. 이 main() 메소드 내부에서 GetHostInfo 및 CallConfigUtil 클래스 객체를 선언하고 호출하여 구동시킨다.

GetHostInfo 클래스는 /proc 파일 체계를 이용해 호스트 상태정보를 읽어들이는 역할을 한다. 성능 감시를 위해 CPU 정보, Memory 정보, SWAP 정보, 평균 부하 정보 및 프로세스와 네트워크에 관한 정보들을 수집한다. 이 정보들은 각각 /proc 디렉토리에 존재하는 cpuinfo, meminfo, loadavg,

\$PID/status, net/dev 등의 파일에서 획득하게 되며 획득한 정보는 추출되어 일정 시간마다 관리자 노드로 전송되어 출력된다.

마지막으로 CallConfigUtil 클래스는 관리자 노드의 사용자가 SAN 노드 설정을 위한 요청을 할 때 사용자 요구를 받아들여 서버 노드의 설정 작업을 수행해주는 일을 담당한다. 이 클래스 역시 소켓 연결을 통해 통신을 수행하며, 소켓을 통해 노드 제어 명령을 송수신한다. 요청된 작업 명령은 CallConfigUtil내부에 구현된 명령어별 해당 작업 유틸리티를 호출하여 실제 설정 작업을 수행한다.

호스트 측 수행 프로그램은 "HostDaemon.java" 라는 단일 자바 파일로 이루어져있으며, 내부적으로 위에서 언급한 작업 처리를 위한 개별 클래스들과 메소드들이 구현되어있다. 따라서, SANtopia 노드 컴퓨터상의 작업 처리를 위해서는 "HostDaemon.java" 파일을 컴파일 한 후 이 파일 하나만 실행을 시켜주면 호스트 측 작업은 모두 끝이 난다.

■ 관리자 소프트웨어 모듈

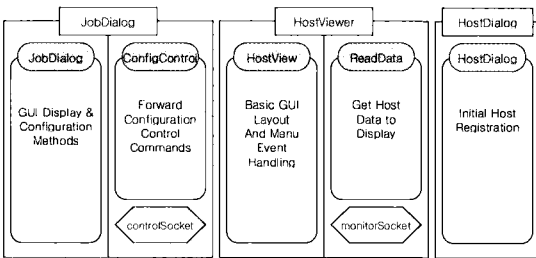
관리자 소프트웨어 모듈은 SANtopia 시스템을 모니터링하고 구성 관리를 하기 위한 관리자용 컴퓨터 노드에서 구동되어지며, GUI 환경을 제공한다.

임무는 다음과 같다.

- 호스트 관련 기본 초기화 정보 입력
- 그래픽 사용자 인터페이스를 제공
- 사용자 인터페이스에서 사용자의 동작에 따른 이벤트를 처리
- 호스트 측 프로그램과의 통신 제공

관리자 노드 측의 프로그램은 그림 7에 나타낸 바와 같이 "HostViewer.java", "JobDialog.java", "HostDialog.java" 의 3개 자바 파일로 구성되어

있으며, 이들은 기본적으로 각각 그래픽 사용자 인터페이스의 제공 및 통신 담당, 사용자 발생 이벤트에 따른 유틸리티 수행 명령 전달, SAN 노드 시스템의 초기 등록을 담당하고 있다. 물론, 파일의 개수는 3개이지만, 이 3개의 파일 안에는 여러 클래스와 메소드들이 구현되어 관련 작업들을 담당한다. 관리자 소프트웨어 모듈은 HostView, ReadData, JobDialog, ConfigControl, HostDialog 등의 5개의 기본적인 클래스들로 이루어져 있다. 이 가운데 HostView 클래스가 기본 클래스로서 프로그램의 수행은 여기서부터 시작된다. 그 외의 다른 클래스들은 HostView 클래스에서 시작되는 작업 이벤트의 처리 및 GUI 창 구현, 호스트 노드와의 정보 및 제어 명령의 송수신 등을 담당하며 클라이언트 프로그램의 세부 기능을 구현한다. 이들 개개의 클래스들에서 처리되는 작업들이 HostView 클래스에의해 조직화되고 관리된다.



(그림 7) 관리자 소프트웨어 모듈의 클래스 구성

HostView 클래스는 관리자 노드 상에서 GUI 사용자 작업환경을 구현해주며, 프로그램 사용을 위한 관련 메뉴 및 그에 따른 이벤트 처리를 담당해 준다. ReadData 클래스는 호스트와 소켓 접속을 시도하여, 호스트로부터 시스템 정보를 얻어오는 일을 담당한다. 사용자 화면 출력을 담당하므로 HostView 클래스와 상호작용을 한다. JobDialog 클래스는 HostView 클래스에 의한 사용자 GUI 환경에서, 관리 및 설정 메뉴의 동작 및 처리를 담당한다. 사용자가 관리 메뉴나 명령을 선택할 경우, JobDialog 클래스에서 그와 관련된 GUI 창의

생성이나, 명령에 따른 직접적인 이벤트의 처리를 담당한다. ConfigControl 클래스는 JobDialog 클래스에 의해 처리되는 명령을 호스트에 전달해 주기 위해 사용되며, 이를 위해 ReadData 클래스와는 별도의 소켓 통신을 이용한다. JobDialog 클래스에서 처리하는 명령을 실제 호스트로 전송하여 처리해주는 일을 담당한다. HostDialog 클래스는 관리자 노드 측 프로그램이 처음 수행된 후, SAN 호스트 노드들을 등록해주는 일을 담당한다. SAN 노드 등록을 위한 GUI 창의 생성 및 입력 정보를 저장하고 처리해주는 일련의 일들을 담당하게 된다. HostView 클래스는 실제 디스크 정보 표시를 위한 Disk 클래스, 각 호스트 정보를 저장하며 갱신하는 Host 클래스, 네트워크 관련 정보를 수집하는 Net 클래스, 프로세스 테이블용 ProcessTableModel 클래스 등과 연결된다.

"JobDialog" 파일은 JobDialog 클래스와 ConfigControl 클래스의 2개의 클래스로 이루어져 있다. JobDialog 클래스는 Configuration 탭의 각 설정 메뉴들을 사용자가 선택할 경우, 작업에 필요한 별도의 작업 창들을 수행하는 부분이다. HostView 클래스에서 구현되는 주 작업 창과는 별도로, 시스템 설정을 위한 작업 창을 생성하고 사용자에게 제공해주게 된다. 이렇게 별도의 파일로써 설정용 GUI 부분을 구현함으로써, 소스 코드가 보다 간결해지고 작업별로 파일이 구분되어 있으므로 차후 수정이나 관리가 용이하도록 했다.

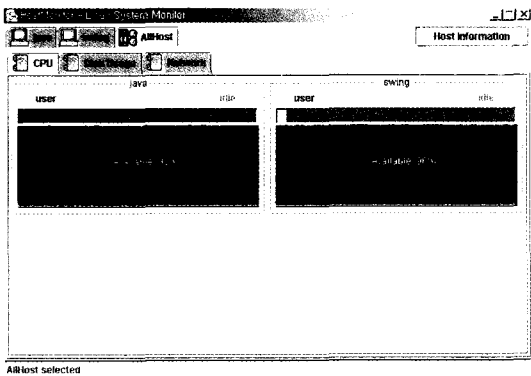
이들을 구현하기 위해 자바 SWING 라이브러리를 이용하므로 GUI 패키지를 포함하며, 클래스의 통신을 위해 네트워크 관련 패키지를 포함한다.

5. 구현 결과

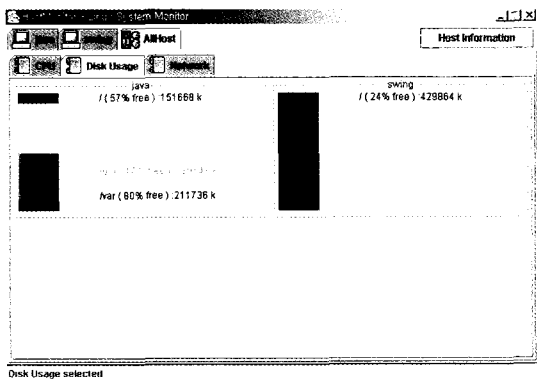
본 관리 도구는 자바 언어를 이용하여 작성되었으며, 시스템 정보의 획득을 위해 리눅스 /proc 파일 체계를 이용하였다. /proc 파일 체계의 이용으로 보다 많은 정보를 보다 쉽게 얻어낼 수 있으며

자바 언어의 사용으로 이중 플랫폼으로의 용이한 이식성을 획득하였다. 그래픽 사용자 인터페이스를 위해서 자바의 SWING 그래픽 라이브러리를 이용했으며, 사용자 동작에 따른 발생 이벤트의 처리는 자바의 내부 이벤트 처리 인터페이스를 사용했고, 서버 노드와의 데이터 및 제어 명령을 위한 통신은 TCP 소켓을 사용해 구현했다.

기본적으로 관리자 프로그램은 탭 구조를 가진다. 크게 보면 시스템의 상태를 보여주는 부분과 구성 관리를 위한 부분으로 나눌 수 있다. 그림 8은 SANTopia 시스템 내의 모든 호스트들의 CPU 사용량, MEMORY 사용량, SWAP 사용량, 부하 정보 중에서 CPU 사용량을 나타낸 예이다. 이것을 보면 한눈에 각 호스트 별 부하를 확인할 수 있다.

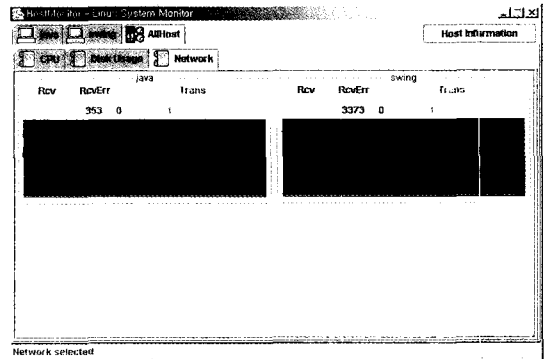


(그림 8) 호스트들의 CPU 상태 표시 예



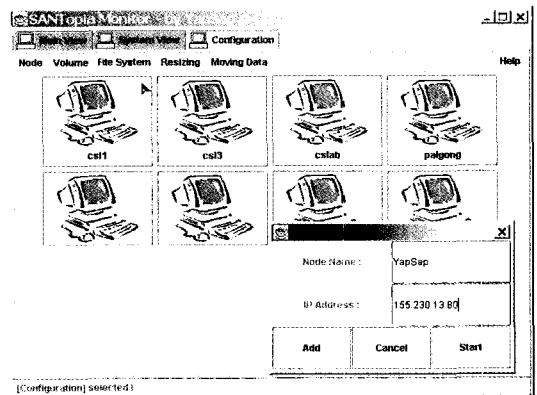
(그림 9) 호스트별로 등록된 디스크 사용량 표시 예

그림 9는 각 호스트에 등록된 디스크 공간의 사용 상태를 나타낸 것이다. 호스트 별로 시스템이 사용하는 저장 공간, 일반 사용자가 점유한 저장 공간, 사용되지 않은 저장 공간의 크기를 알 수 있다. 그림 10은 각 호스트 통신량을 나타낸 것이다. 호스트별로 송수신한 데이터의 량, 전송 상의 오류 발생여부를 알 수 있다.



(그림 10) 전체 호스트의 네트워크 상태 표시 예

호스트가 관리자와 연결된 상태에서 Configuration 탭을 선택했을 때의 화면 구성은 그림 11과 같다.



(그림 11) 구성 관리 화면 예

이곳에서는 노드 설정에 관련된 여러 작업들이 이루어진다. Node, Volume, File System, Resizing, Moving Data 등의 메뉴를 선택하면, 각

항목과 관련된 하위 메뉴들이 풀다운 형식으로 나타난다. 메뉴 창 아래의 영역에는 현재 등록된 서버 노드들이 이름과 함께 아이콘 형식으로 표시된다. 이를 보고 관리자는 현재 등록된 노드가 어떤 것이 있는지 쉽게 알 수 있다.

메뉴 구현 부는 주 작업 창과는 별도의 팝업 창을 구현하여, 사용자 명령을 입력받는다. 팝업 창들은 주 작업 창과는 별도의 프레임을 할당해 구현했다.

본 구성 관리 도구는 SANtopia 시스템에서 노드의 생성이나 삭제, 파일 시스템의 생성 및 제거 등과 같은 SANtopia 시스템 전체 환경을 설정하고 관리해 주는 작업을 수행하게 된다.

6. 결 론

SAN(Storage Area Network)이 저장장치의 비용절감과 유연한 시스템 구성과 같은 많은 장점을 가진 반면 기존의 저장 환경보다는 매우 복잡해졌다. SAN과 함께 발생한 복잡성은 새로운 저장 관리의 필요성을 요구하게 되었다. 파이버 채널 기술을 이용하는 SAN환경에 있어서는 수백 개의 장치들이 시스템을 구성하게 되므로 효율적인 관리를 위한 새로운 관리 서비스와 정책이 제시될 필요가 있다.

본 논문이 기여하는 바는 크게 두 가지로 볼 수 있다. 우선 SAN 관리 소프트웨어들이 다양하게 연구 개발되고 있는데 이전 연구를 바탕으로 네트워크로 연결된 저장 관리에 필요한 일반적인 관리 기능들을 계층 프레임워크로 정의해 보았다는 것이다. 두 번째는 실제 시스템인 한국전자통신연구원(ETRI)의 SANtopia 시스템의 성능 감시 및 구성 관리 도구 개발 프로토타입을 제시하였다는 것이다. 이 도구의 차별성은 이식성과 용이성이다. 이 도구는 자바 언어를 이용한 GUI 기반으로 개발되었다. 크게 호스트와 관리 노드 부분으로 나뉘어져 동작하게 되며, 각 호스트 상에서 데몬 형식으로

구동되어 사용자가 요청하는 작업을 수행하게 되며, 관리자 컴퓨터에서 구동되어 SANtopia 시스템의 상태를 모니터링하고 필요한 설정을 가능케 하는 GUI 작업 환경을 제공한다.

기본적으로 SANtopia 시스템은 리눅스를 기반으로 하는 SAN 시스템임을 가정하고 있으나, 본 도구는 자바 언어를 기반으로 하고 있어 유닉스, 윈도우, 솔라리스 등 다른 플랫폼 환경에도 필요한 경우에는 쉽게 이식이 가능하여 이 기종들로 구성된 시스템의 운영 시에도 쉽게 적용이 가능하다. 또한, 일반적인 유닉스/리눅스 계열의 운영체제에서는 시스템 환경 설정 등의 작업들이 명령어 입력 방식의 텍스트 기반으로 되어 있어 사용과 관리에 있어 불편한 점이 많았다. 이에 본 관리 시스템에서는 자바 SWING 라이브러리를 사용한 GUI 기반의 작업 환경을 제공하여 보다 쉽고 편리한, 효율적인 시스템 관리가 가능하다.

향후 다른 플랫폼 상의 SAN 시스템을 위한 관리 도구로서의 이식 및 적용이 고려될 수 있을 것이며 그 경우 자바로 구현된 사용자 환경은 변화 없이 각 시스템에 적적할 하부 계층을 연결시켜 주기만 하면 될 것이다. 하지만 상대적으로 느린 자바 언어의 사용 시 발생하는 속도 문제 개선을 위한 최적화가 필요하겠다.

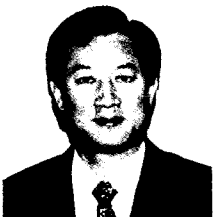
참 고 문 헌

- [1] 민병준, 임기욱, "네트워크 연결형 저장시스템 기술", 정보통신연구진흥지, 2000.
- [2] 박성순, 박명순, 신해선, 천근영, 백현기, "SAN 관리 소프트웨어 기술, 그 제품 및 전망", 정보과학회지 2001년 3월 제19권 제3호 통권 제142호.
- [3] 서대화, 민병준, 임기욱, "네트워크 연결형 스토리지의 기술동향", 정보과학회지 2001년 3월 제19권 제3호 통권 제142호.
- [4] Benner, "Fibre Channel: Gigabit Communications and I/O for Computer

- Network", McGraw-Hill, 1996.
- [5] Fibre Channel Association, "Fibre Channel Solutions : Management", www.fibre channel.com/, 1999.
- [6] Wilson, "Managing a Fibre Channel Storage Area Network", White Paper, Storage Network Industry Association, 1999, www.snia.org.
- [7] Sun Microsystems, "Sun's Managed Storage Networks", White Paper, 1999.
- [8] Schulz, "Solving Storage Management Challenges with Storage Area Networks (SAN)", White Paper, MTI Corporate, 1998.
- [9] Compaq, "Compaq SANworks Enterprise Network Storage Management", 2000, www.compaq.com.
- [10] Kenneth W. et al., A 64-bit, Shared Disk File System for Linux, In The 7th NASA Goddard Conference on Mass Storage System and Technologies in cooperation with the 16th IEEE Symposium on Mass Storage Systems, pp.22-41, San Diego, USA, March 1999.
- [11] Clark, "A Management Strategy for Fibre Channel Arbitrated Loop", White Paper, Vixel Corporation, www.vixel.com, 1998.
- [12] HighGround systems, "How to Plan for Storage Area Network(SAN)", White Paper, www.highground.com, 2000.
- [13] Jensen, "Managing the SAN", White paper, 1999, Gadzoox Network Inc., www.gadzoox.com.
- [14] Oxenhandler, Tom Clark, " Managing Multivendor SANs with SAN InSite 2000", White Paper, Vixel corporation, 2000, www.vixel.com.
- [15] 김경배, 김영호, 김창수, 신범주, "SAN을 위한 전역 파일 공유 시스템의 개발", 정보과학회지 2001년 3월 제19권 제3호 통권 제 142호

● 저 자 소 개 ●

임 기 옥



1977년 인하대학교 공과대학 전자공학(학사)
 1987년 한양대학교 전자계산학(석사)
 1994년 인하대학교 전자계산학(박사)
 1977~1983 한국전자기술연구소 선임연구원
 1983~1988 한국전자통신연구소 시스템소프트웨어 연구실장
 1988~1989 미 캘리포니아 주립대학(Irvine) 방문연구원
 1989~1997 한국전자통신연구원 시스템연구부장, 주전산기(타이컴) III, IV 개발 사업책임자
 1997~1999 정보통신연구진흥원 정보기술전문위원
 2001~2003 ETRI 컴퓨터소프트웨어연구소소장
 2000~현재 : 선문대학교 지식정보산업공학과 교수
 관심분야 : 실시간 데이터베이스시스템, 시스템구조, 운영체제
 E-Mail: rim@sunmoon.ac.kr

● 저 자 소개 ●



나 용 희

2002년 인천대학교 컴퓨터 공학과 졸업(학사)
2002년~현재 : 인천대학교 컴퓨터공학과 (석사과정)
관심분야 : Intrusion Tolerant System, Active Network
E-mail : ramen@incheon.ac.kr



민 병 준

1983년 연세대학교 전자공학과(학사)
1985년 연세대학교 대학원 전자공학과 (석사)
1991년 미국 캘리포니아주립대학교(UC Irvine) 컴퓨터공학과 (박사)
1995년~현재 : 인천대학교 컴퓨터공학과 부교수
관심분야 : 분산 시스템, 시스템보안, 망 관리
E-mail : bjmin@incheon.ac.kr



서 대 화

1981년 경북대학교 전자공학과졸업(학사)
1983년 한국과학기술원 전산학과 졸업(석사)
1993년 한국과학기술원 전산학과 졸업(박사)
1993년~1995년 한국전자통신연구소
1995년~현재 : 경북대학교 전자전기공학부 교수
관심분야 : 분산운영체제, 임베디드 리눅스, 이동컴퓨팅 구조
E-mail : dwseo@ee.knu.ac.kr



신 범 주

1983년 경북대학교 전자공학과(학사)
1991년 경북대학교 컴퓨터공학과(석사)
1998년 경북대학교 컴퓨터공학과(박사)
1987년~2002년 한국전자통신연구원(책임연구원)
2002년~현재 : 밀양대학교 컴퓨터공학과 교수
관심분야 : 고장감내 분산시스템, 스토리지 클러스터 소프트웨어, 분산 임베디드 시스템 등
E-Mail: bjshin@mnu.ac.kr