

리눅스 클러스터 시스템 통합 관리 도구

(Integrated Linux Cluster System Administration Tool)

김은희[†] 김지연^{**} 박용관[†] 권성주[†] 최재영^{***}

(Eunhoe Kim) (Jiyeon Kim) (Yongkwan Park) (Sungju Kwon) (Jaeyoung Choi)

요약 본 논문에서는 리눅스 클러스터 시스템 통합 관리 도구인 CATS-i(Cluster Administration Tools on the Internet)의 시스템 구성과 디자인 문제들을 논한다. CATS-i는 리눅스 클러스터 시스템을 쉽고 빠르고 안전하게 설치하고 관리하기 위하여 개발된 도구이다. 클러스터의 운영체제 설치에서부터 응용 프로그램 패키지 설치, 클러스터 노드들의 자원들을 실시간으로 모니터링하고 관리할 수 있는 기능, 배치 작업 제출 및 관리 기능들이 통합되어 사용자에게 단일 시스템 이미지를 제공한다. 또한 사용자가 플랫폼에 관계없이 쉽고 편리하게 클러스터의 상태를 파악하고 관리할 수 있는 자바 기반의 강력한 그래픽 사용자 인터페이스를 제공한다.

키워드 : 클러스터, 클러스터 관리 시스템, 병렬 컴퓨팅, 고성능 컴퓨팅

Abstract In this paper, we discuss the system configuration and the design issues of CATS-i, a set of installation and administration tools for Linux cluster systems. CATS-i enables users to manage cluster systems easily, quickly, and safely. It integrates many functions, ranging from installing operating systems and applications to real-time monitoring and management of various important resources. In addition, batch job submission and management are included. These functions support a single system image. Finally, a powerful graphic user interface based on Java lets users quickly understand the cluster status and conveniently use the CATS-i on the Web.

Key words : Cluster system, CMS, Parallel Computing, High Performance Computing

1. 서론

오늘날 인터넷 중심의 컴퓨팅 환경에 적합한 시스템은 적은 비용으로 고성능, 고가용성, 그리고 강력한 확장성을 제공하는 시스템이다[1]. 현재 이러한 요구조건을 만족하는 솔루션으로 클러스터 시스템이 각광받고 있다. 클러스터 시스템은 두 대 이상의 컴퓨터를 서로 연결하여 하나의 커다란 단일 시스템 환경을 제공한다. 1994년 Thomas Sterling과 Don Becker가 16개의 DX2 프로세서를 연결하여 Beowulf라는 클러스터 시스

템을 구축하면서 주목받기 시작하였다[2]. 이후 클러스터를 구성하는 컴퓨터 노드의 수는 수십 대에서 수백 대 이상으로 점점 규모가 커지고 있다. 클러스터의 규모가 커질수록 관리의 어려움 또한 비례하여 증가한다[3]. 클러스터 시스템의 하드웨어는 일단 구성하고 나면 관리할 필요가 적지만 운영체제와 기타 프로그램은 용도에 따라 설치, 업그레이드, 삭제하는 작업을 끊임없이 해야하기 때문이다. 이외에도 자원 모니터링과 계정, 서버 시스템 관리, 네트워크 관리, 보안, 에러 보고와 같은 문제 처리 등의 작업들을 지속적으로 해야 한다. 클러스터 관리 작업을 관리와 모니터링 주제로 분류하여 정리하면 표 1과 같다[1, 4].

기존의 클러스터 관리 도구들 중에는 워크스테이션 전용으로 개발되었던 모니터링 도구를 확장한 것이 많다. 이러한 관리 도구들은 기능이 강력하지만 클러스터 시스템 전반에 대한 감시와 관리보다는 네트워크 자원 중심의 모니터링 도구이다[2]. 한편 클러스터 전용으로 개발된 관리 도구들도 16개 이상의 노드들로 구성된 대용량 클러스터를 관리하기에는 부적합하다. 왜냐하면 대

· 이 논문은 교육부 BK-21 핵심분야 지원사업과 자이온리눅스시스템즈(주)의 지원으로 수행되었습니다.

[†] 비회원 : 숭실대학교 컴퓨터학부
ehkim@ss.ssu.ac.kr
psiver@ss.ssu.ac.kr
lithmoon@ss.ssu.ac.kr

^{**} 비회원 : 한국전자통신연구원 컴퓨터소프트웨어연구소 연구원
jiyeonkim@etri.re.kr

^{***} 종신회원 : 숭실대학교 컴퓨터학부 교수
choi@comp.ssu.ac.kr

논문접수 : 2002년 5월 27일

심사완료 : 2002년 8월 5일

표 1 클러스터 관리 작업 분류

	분류	세부 항목
관 리	자원 관리	운영체제 설치 및 업그레이드, 프로그램 설치, 업그레이드, 삭제 디바이스, 구성, 사용자, 계정
	서브 시스템 관리	서브 시스템 구성, 제어
	네트워크 관리	알람 기능, SNMP, NetView
	보 안	C2, 사용자 인증
	문제 처리	알람 기능, 자가 진단, 에러 리포트, 로깅
모 니 터	물리적인 환경	CPU 온도, 습도, 팬의 속도
	논리적인 서비스	프로세스 가용성
	Performance Meters	CPU, 메모리, 디스크 사용률, 네트워크 트래픽, I/O 병목

부분의 클러스터 관리 도구들은 소규모 클러스터를 관리하기 위한 맞춤형 도구이므로 확장성을 제공하지 못하기 때문이다[5]. 또한 대부분의 클러스터 관리 도구들은 관리자의 전용물로서 클러스터 사용자에게 대한 지원이 없다. 클러스터 사용자도 제한된 사용 권한을 가지고 클러스터 상태를 파악할 수 있도록 서비스를 지원해야 한다[1].

CATS-*i*는 Cluster Administration ToolS on the Internet의 약자로 인터넷에서 리눅스 클러스터 시스템을 쉽고 빠르고 안전하게 설치하고 관리하기 위하여 개발된 도구이다. 클러스터의 운영체제 설치에서부터 응용 프로그램 설치, 자원들의 실시간 모니터링 및 관리, 네트워크 모니터링, 파일 관리, 자원 활용의 통계, 배치 작업 제출 및 관리 등의 기능들이 통합되어 사용자에게 단일 시스템 이미지를 제공한다. CATS-*i*는 대용량 클러스터 시스템을 관리하기 위해 쉽게 확장이 가능하다. 사용자 인터페이스는 자바로 구현되었기 때문에 플랫폼에 독립적이며, 웹으로 클러스터를 관리할 수 있다. 그리고 관리자뿐만 아니라 일반 사용자를 위한 서비스를 지원하는 클러스터 관리 도구이다.

본 논문의 구성은 다음과 같다. 2장에서는 CATS-*i*의 전체 시스템을 설명한다. 특히 각각의 서브 시스템을 중심으로 시스템 구성과 디자인 초점을 논한다. 3장은 CATS-*i* 구현과 성능평가를 실험 결과를 토대로 분석한다. 4장에서는 관련 연구들과 CATS-*i*를 비교 설명한다. 5장에서는 결론 및 향후 연구방향을 제시한다.

2. CATS-*i*

CATS-*i*는 3-tier 구조로 서버 데몬, 노드 데몬, 그리

고 사용자 인터페이스로 구성된다. 그림 1은 CATS-*i*의 전체 시스템 구성이다. 서버 데몬으로는 운영체제 설치 데몬(sender_inx, progd), 프로그램 설치 데몬(applcid), 클러스터 모니터 및 관리 데몬(sAIID), 알람 데몬(AlarmServer), 인증 데몬(AuthServer), 스위치 트래픽 모니터 데몬(NetMonServer), 배치 작업 제출 및 관리 데몬(JPBS)이 있다. 서버 노드에서는 사용자 인터페이스로부터 서비스 요청을 받아 사용자 인증을 확인하고, 운영체제 설치와 프로그램 설치 요청을 받아 이를 실행한다. 또한 노드 데몬으로부터 자원들의 모니터링 결과를 받아 사용자 인터페이스에 전달하고 자원에 대한 한계 상황이 발생했을 경우 알람 기능으로 관리자에게 알리고, 기타 스위치 모니터와 파일 관리 등과 같은 서비스를 제공한다. 노드 데몬은 실제 모니터와 자원 관리의 대상이 되는 클러스터의 모든 노드에서 실행된다. 서버 노드의 sAIID의 요청에 따라 노드의 자원 모니터링과 관리를 담당하는 cNodeD와 각 노드의 프로그램 설치를 담당하는 appdaemon으로 구성된다. 사용자 인터페이스는 프로그램 또는 웹으로 실행되는 GUI이다.

2.1 운영체제 설치

클러스터의 모든 노드에 각각 운영체제를 설치하고 설정하는 것은 시간이 많이 소비되는 반복적인 작업이다. 기존에는 NFS(Network File System)를 이용하여 리모트로 클러스터 노드에 운영체제를 설치하는 방법이 많이 사용되었다[6]. 이 방법은 적은 노드를 가진 클러스터에서 NFS의 안정성 때문에 빠르고 안정적으로 운영체제를 설치할 수 있다. 하지만 많은 노드들로 구성되어 있는 클러스터에서 운영체제를 설치할 때는 서버 노드에 입출력 부하가 크게 증가하기 때문에, 대용량 클러스터의 운영체제 설치 방법으로는 부적합하다.

CATS-*i*에서는 멀티캐스트 이미지 전송을 사용한 운영체제 설치 방법을 개발하여 사용한다. 멀티캐스트 이

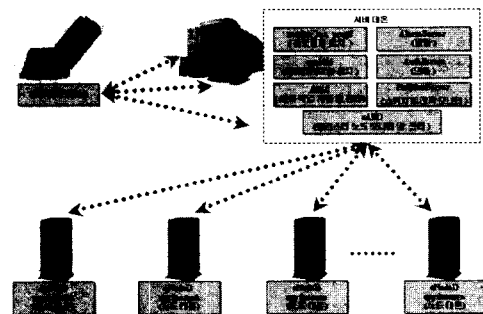


그림 1 CATS-*i* 시스템 구성

이미지 전송을 사용한 운영체제 설치의 먼저 시스템 이미지를 압축하여 서버 노드에 위치시키고, 운영체제를 설치할 노드는 부팅 디스켓이나 Bootp와 tftp를 이용한 네트워크 부트를 사용해 부팅시킨다. NFS를 통해서 멀티캐스트에 이용될 설치 모듈을 운영체제가 설치될 모든 노드의 램드라이브에 설치한 후, 인터넷 주소의 멀티캐스트 주소 영역인 D 클래스 주소를 (244.0.0.0 ~ 239.255.255.255) 할당하고 UDP 기반으로 서버의 이미지를 멀티캐스팅하여 운영체제를 설치한다. UDP는 TCP 기반의 통신보다 속도가 빠르다는 장점이 있지만 메시지 손실이 있을 수 있으므로 별도의 에러/흐름 제어 모듈로 통신의 신뢰성을 보장한다. 그림 2는 멀티캐스트 이미지 전송 방법을 사용한 운영체제 설치 시스템의 구성을 보여준다. sender_lnx 데몬은 멀티캐스트 이미지 전송을 통한 운영체제 설치 전반을 담당하며, progfd는 설치 진행 상황을 체크하여 알려주는 데몬이다.

CATS-*i*의 멀티캐스트 이미지 전송을 이용한 운영체제 설치 도구는 멀티캐스트를 사용하므로 동시에 클러스터의 모든 노드에 운영체제를 쉽고 빠르게 설치하거나 갱신할 수 있다. 많은 노드를 가진 클러스터에서도 NFS를 사용했을 때의 입출력 부하의 과도한 증가는 발생하지 않으므로 대용량 클러스터 설치에 적합하다.

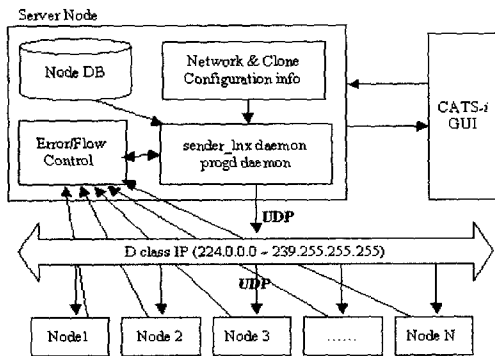


그림 2 멀티캐스트 이미지 전송을 사용한 운영체제 설치 시스템 구성

2.2 프로그램 설치

CATS-*i*는 프로그램의 설치, 삭제, 갱신뿐 아니라 프로그램이 설치되었는지를 확인하는 기능을 지원한다. 프로그램 설치 시스템은 RPM 장치 라이브러리 시스템 호출을 이용하므로 레드햇 기반의 리눅스 시스템에서 잘 작동된다. 그림 3과 같이 사용자의 요청을 받아서 서버 노드의 appclid 데몬이 프로그램을 설치할 노드로

RPM 파일을 전송한다. 프로그램을 설치할 노드의 appdaemon 데몬은 서버에서 전송한 RPM 파일을 전송 받고, RPM 시스템 호출을 통해 실제로 프로그램을 설치한다. appdaemon은 프로그램 설치 작업 외에도 각 노드의 프로그램 설치 정보를 서비스한다.

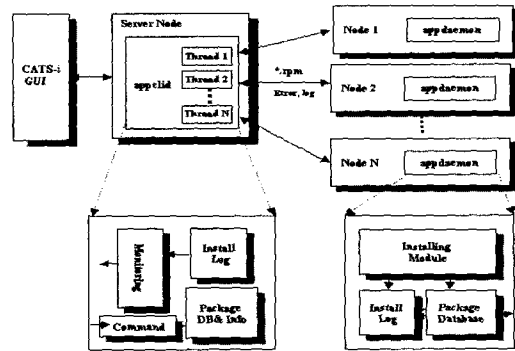


그림 3 프로그램 설치 시스템 구성

2.3 클러스터 노드의 모니터와 관리

그림 1에서 보듯이 사용자가 사용자 인터페이스에서 클러스터 노드의 모니터와 관리 서비스를 요청하면, sAIIID 데몬은 해당 서비스를 파악하여 필요에 따라 각 노드의 cNodeD 데몬에 정보를 요청한다. sAIIID는 cNodeD로부터 요청한 정보를 받아 사용자 인터페이스로 정보를 보내는 작업 흐름을 가진다. 클러스터의 각 노드별로 모니터 또는 관리되는 자원 및 기타 항목은 표 2와 같다. 표 2에서 나타나듯이 CATS-*i*는 클러스터 노드의 다양한 자원들을 모니터링하고 관리할 수 있는 도구이다. CATS-*i*는 다음과 같은 특징을 가지고 있다.

- a) 클러스터의 노드들을 사용 목적이나 도메인 등에 따라서 논리적인 그룹으로 묶을 수 있다. 따라서 CATS-*i*는 노드 단위, 클러스터 그룹 단위, 전체 클러스터 단위로 클러스터를 모니터 및 관리할 수 있으므로 사용자가 클러스터 도구를 더 효율적으로 사용할 수 있다.
- b) 자원들에 대한 실시간 모니터링을 제공한다. CPU, 디스크, 메모리, 마더보드, 프로세스 등과 같은 중요한 시스템 자원들에 대하여 실시간 모니터링을 제공한다. 모니터의 융통성을 위해 사용자가 작업 환경에 적합하게 모니터 주기를 설정할 수 있다.
- c) 클러스터 전체 또는 그룹 단위로 동일한 명령을 동시에 실행시킬 수 있다. 이러한 병렬 명령어 처리 기능은 클러스터의 관리 시간을 단축시켜 준다.

표 2 클러스터 노드별 모니터 및 관리 항목

항 목	설 명	시행 항목
CPU	CPU 사용률 모니터	User · Nice · System별 CPU 사용률
Disk	하드 디스크 사용률 모니터	디스크 사용률, 마운트 포인트, 디스크 전체 크기, 사용된 디스크 크기, 사용되지 않은 디스크 크기, 사용중인 파일시스템 종류
Memory	메모리 사용률 모니터	프로그램, 버퍼, 캐쉬, 스왑별 메모리 사용률
Node status, Information	노드의 활성화 정보 및 기본적인 정보 제공	운영체제 종류, 커널 버전, 기종, CPU 모델명, CPU 클럭 속도, RAM 크기
Node Control	노드의 제어	셋다운, 리부팅, 리모트 전원 켜기
User	노드에 접속해 있는 사용자 모니터	사용자 아이디, 터미널, 로그인 시간
Account	계정 관리	계정 정보, 계정 등록 · 삭제 · 수정
Process	프로세스 모니터링 및 관리	프로세스 이름 · PID · 유저 · 상태 · 크기 · cputime · Nice · 시작 명령어, 프로세스 종료 · 정지 · 재개
Motherboard	머더 보드 모니터	보드의 온도, CPU 온도, 팬 속도
Scheduling Commands	명령어 스케줄링	Cron 테이블 생성, 편집, 삭제
File	파일 시스템 관리	파일시스템 트리 뷰, 파일 및 디렉토리 삭제 · 생성 · 변경, 파일 및 디렉토리 이동 · 복사, 파일 편집 및 속성 변경
Parallel Command	모든 노드에 동일한 명령어를 동시에 실행	pcat, prm, pexec, pKillUser, pKillPName, pcp, pmv
Log	시스템 로그 관리	로그 트리 뷰, 날짜별 주제별 검색, 등록, 저장, 명세, 등급 설정

- d) 강력한 파일 관리 기능을 가지고 있다. 윈도우즈 운영체제의 탐색기 인터페이스와 유사한 인터페이스를 제공한다. 따라서 사용자들은 파일 또는 디렉토리의 생성 · 삭제 · 변경뿐만 아니라 노드간 파일 및 디렉토리의 복사와 이동, 파일 편집 등의 작업을 사용자가 쉽게 처리할 수 있다.
- e) 사용자 및 노드 단위로 자원의 사용률을 통계하여 보여준다. 즉, 어느 자원을 어떤 사용자가 얼마큼 사용하였는지, 또는 어느 노드의 시스템 로드가 얼마나 걸리는지에 대한 정보를 쉽게 파악할 수 있다.

2.4 스위치 트래픽 모니터

클러스터 노드들은 보통 고성능 스위치의 네트워크 인터페이스에 연결된다. 스위치를 사용하는 네트워크는 스위치의 네트워크 인터페이스들을 기점으로 서로 다른 충돌 도메인(Collision Domain)을 가지므로 네트워크 트래픽을 모니터하기 위해서는 클러스터내 모든 노드들에 각각 별도의 데몬을 실행시켜야 한다. 따라서 네트워크 트래픽 모니터를 위한 클러스터 시스템의 부하가 증가하게 되며, 클러스터의 사용목적이 인터넷 서비스를 위한 서버일 때는 네트워크의 사용이 크게 증가하므

로 시스템 성능이 감소한다[7]. 한편 클러스터의 노드간 통신은 스위치를 거쳐서 이루어지고 있으며, 외부와의 통신 또한 스위치를 통해서 이루어진다. 따라서 스위치의 활성화된 네트워크 인터페이스에 대한 트래픽 정보는 곧 클러스터의 네트워크 상태와 부하 정보라는 특성이 있다[8, 9]. CATS-i에서는 네트워크 모니터링을 위한 클러스터 시스템의 부하를 최소화하고 클러스터 네트워크의 대역폭 사용률과 상태를 파악할 수 있는 스위치 트래픽 모니터 시스템을 사용한다.

그림 4는 스위치 트래픽 모니터 시스템의 구조를 보여주고 있다. NetMonServer는 NetMon 쓰레드를 생성하고 사용자의 요청을 서비스한다. 생성된 NetMon 쓰레드는 주기적으로 SNMP 에이전트가 탑재되어 있는 스위치의 트래픽 정보를 SNMP(Simple Network Management Protocol)를 사용하여 가져온 후, 데이터 저장소에 저장한다. 트래픽 정보의 종류는 스위치의 모든 활성화된 인터페이스 각각에 들어오는 트래픽량과 나가는 트래픽량, 그리고 에러 패킷 수 등이며, 이러한 정보를 이용해 각각의 네트워크 인터페이스에 네트워크 대역폭 사용률을 서비스한다.

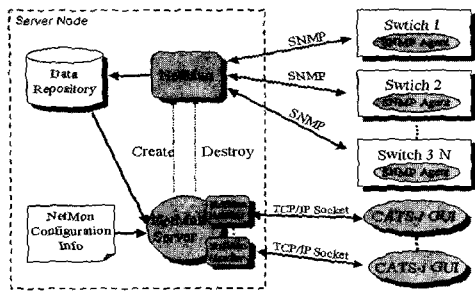


그림 4 스위치 트래픽 모니터 시스템 구성

2.5 알람

알람 데몬은 주기적으로 모든 클러스터 노드의 메모리, CPU, 디스크의 자원 이용률과 로그인 사용자 수, CPU 온도 및 팬 속도를 모니터링해서 관리자가 정한 위험 수치를 넘게 되면 관리자에게 전자우편으로 그 결과를 통보하는 역할을 하는 프로세스이다. 주기적으로 모니터링한 자원의 정보는 데이터 저장소에 저장한다. 관리자는 사용자 인터페이스를 통해 자원들의 위험 수치를 정하고, 알람 데몬은 위험 수치를 따라 전자우편으로 위험성을 알린다. 또한 클러스터 관리자가 클러스터 관리를 위해 사용자에게 메시지를 보낼 수 있는 기능을 지원한다.

2.6 인증

인증 데몬은 서버 노드에 수행되는 프로세스로 CATS-i의 로그인 사용자 인증을 담당한다. 사용자 인증의 보안을 위해 SSL(Secure Socket Layer)과 TLS (Transport Layer Security) 계층의 보안을 제공하는 JSSE[10]를 사용한다. 인증 데몬은 사용자의 아이디와 패스워드를 수신하여 리눅스 파일 시스템내의 /etc/passwd 파일 정보와 비교하며 사용자의 아이디와 패스워드를 확인하는 인증 절차를 가진다. 또한 CATS-i는 관리자와 일반 사용자에게 서로 다른 사용 권한을 부여함으로써 일반 사용자도 제한적인 기능으로 클러스터를 모니터링할 수 있다.

2.7 배치 작업 제출 및 관리

JPBS(Java Portable Batch System)는 PBS(Portable Batch System)[11]의 자바 인터페이스로 배치 작업을 제출 및 관리할 수 있다. PBS는 고성능 컴퓨팅을 위해 NASA Ames 연구소에서 개발된 배치 작업 관리 시스템으로 클러스터로의 작업 제출, 멈춤, 재수행, 감시, 삭제 기능뿐만 아니라, 시그널을 통한 작업의 선택, 재수행, 변경 등이 가능하다. PBS는 기본적으로 명령어 기반의 인터페이스만 제공하여 일반 사용자가 사용하기

어려우며, PBS의 사용자 인터페이스를 제공하는 xPBS는 Tcl/Tk로 작성되어 있어 PBS를 사용하기 위해서는 클러스터 콘솔로의 접속이 반드시 필요하다는 단점이 있다[12].

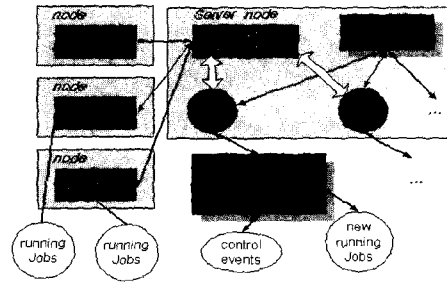


그림 5 JPBS를 사용한 배치 작업 관리 시스템 구성

CATS-i의 JPBS는 이러한 문제점을 해결하여, 클러스터 콘솔로의 접속없이 웹 환경에서도 클러스터에 작업을 제출하고 관리가 가능하도록 지원한다. 그림 5는 JPBS를 이용한 배치 작업 관리 시스템의 전체적인 구성이다. 사용자의 서비스 요청은 CATS-i 사용자 인터페이스를 통하여 JPBS 데몬에게 전달된다. JPBS 데몬은 서비스에 부합하는 적절한 권한을 가진 자식 서버를 생성하여 서비스를 처리한다. PBS는 pbs_server, pbs_scheduler, pbs_mon 데몬들로 구성되며, 생성된 자식 서버는 실제로 PBS 시스템에 서비스 요청을 한다. 또한 자식 서버는 PBS로부터 발생하는 여러 이벤트를 다시 사용자 인터페이스로 전달한다.

3. 구현 및 성능 평가

CATS-i의 서버 데몬 프로그램은 C와 Java로 구현되었다. 특히 클러스터 노드 모니터 및 관리 데몬인 sAID는 다양한 자원들을 실시간 모니터링하여야 하므로 성능을 고려하여 C를 사용하여 구현하였다. 그리고 서버 시스템의 각각의 기능들은 스레드로 동작하도록 구현하였다.

CATS-i의 사용자 인터페이스는 Java로 구현하여 플랫폼에 독립성을 지원하고, 애플릿으로 실행시킬 수 있기 때문에 웹에서 바로 사용할 수 있다. 또한 자바의 스윙을 사용하여 수려한 그래픽과 인터페이스를 지원한다. 클러스터 자원들의 모니터 정보를 사용자가 빨리 쉽게 파악할 수 있도록 막대 그래프, 파이 차트, 선 그래프로 표현한다. 클러스터 그룹과 기타 계층적인 정보를 트리로 표현하여 정보의 직관성을 높였다. 그림 6은 루트 원

도우와 디스크 사용률과 CPU, 메모리 등과 같은 자원의 모니터 윈도우이다. 루트 윈도우의 왼쪽 트리는 클러스터 노드들의 논리적인 그룹을 보여주며, 빨간색 테두

리를 가진 펑귄 아이콘은 응답이 없거나 문제가 발생하여 모니터할 수 없는 노드를 표시한다. 그리고 그림 7에서 (a)는 운영체제 설치, (b)는 프로그램 설치, (c)는 파일 관리, (d)는 스위치 트래픽 모니터, (e)는 JPBS 인터페이스를 보여주고 있다.

CATS-*i*는 3-tier 구조로 되어있으므로 서버 노드에 트래픽이 집중될 수 있다. 특히 클러스터 노드의 자원들을 실시간 모니터 및 관리하는 sAIIID 데몬은 클러스터 노드의 수가 증가할수록 실시간으로 모니터할 자원들이 그에 비례하여 증가하므로 많은 부하가 발생할 수 있다. 클러스터 노드의 수가 30대일 때, sAIIID 데몬을 RAM 128 Mbytes, 600 MHz Pentium III 리눅스 컴퓨터에서 실행시키고, top 유틸리티를 사용하여 CPU, 메모리 사용률을 측정하였다. CATS-*i*의 클러스터 노드

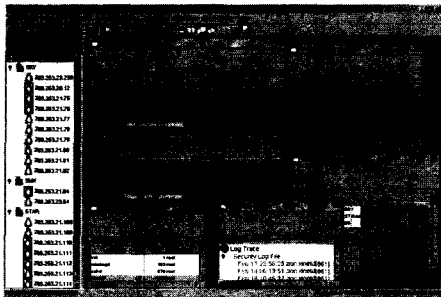
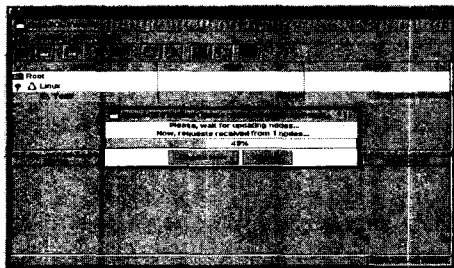
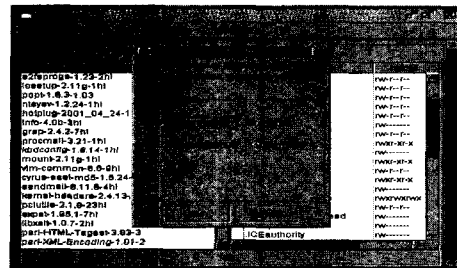


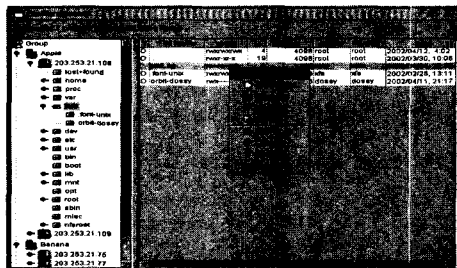
그림 6 CATS-*i* 사용자 인터페이스 루트 윈도우



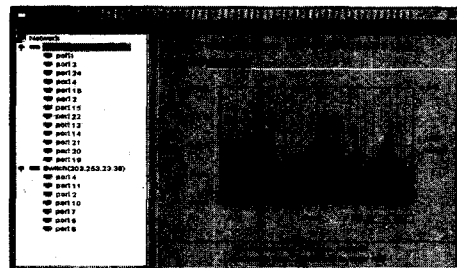
(a) 운영체제 설치 윈도우



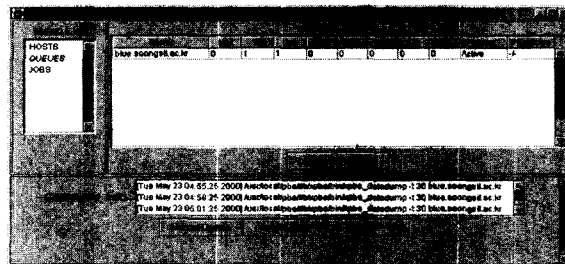
(b) 프로그램 설치 윈도우



(c) 파일 관리 윈도우



(d) 스위치 트래픽 모니터 윈도우



(e) JPBS 메인 윈도우

그림 7 CATS-*i* 사용자 인터페이스

모니터링 기능을 모두 실행시키고, 모니터 주기를 3초로 하였을 때, 평균 10분 동안 3회 반복 측정된 평균값으로 CPU 사용률은 2.0%, 메모리 0.3%를 기록하였다. 클러스터 노드가 30대일 때의 평균값임을 고려할 때, 양호한 성능을 보여주었다.

4. 관련 연구 및 비교

SCMS[13], PARMON[14], GARDMON[15], ClusterProbe[5]는 대표적인 클러스터 관리 도구이다. 이들 중 PARMON, GARDMON은 클라이언트-서버 구조로 모니터링 데이터 수집과 시각화 등의 기능을 모두 클라이언트가 담당한다. 따라서 자원의 실시간 모니터 기능이 강력하고 시스템이 단순하다는 장점이 있다. 하지만 여러 사용자에게 동시에 일관성있는 서비스를 제공하기 어렵고, 클러스터의 논리적인 그룹화를 비롯한 기타 클러스터 관리의 기능이 부족하다는 단점이 있다. CATS-i, SCMS, ClusterProbe는 3-tier 시스템으로 클라이언트-서버 구조의 클러스터 관리 도구가 가지는 단점을 해결한다.

SCMS는 표 1에서 언급한 클러스터 관리 도구의 다양한 기능들을 지원하며, 다양한 병렬처리 명령어를 지원하고, 3차원 그래픽 사용자 인터페이스를 지원하는 특징이 있다. 그러나 C, C++, TCL/TK, Perl, Java와 같은 여러 언어를 사용하여 구현하였기 때문에 시스템이 복잡하고, 설치할 때 패키지 의존성이 많아 설치 과정이 어렵다. GARDMON은 서비스의 가용성 및 제어 기능이 잘 갖춰져 있으며, PARMON은 자원에 대한 자세한 모니터 정보를 제공한다. 하지만 GARDMON, PARMON은 클러스터 자원의 모니터링에 중점을 둔 도구이기 때문에, 관리 기능이 부족하다. ClusterProbe는 Java RMI를 사용하는 계층적인 구조로 관리 시스템을 구성하여 대용량 클러스터를 지원할 수 있다는 특징이 있다.

대표적인 클러스터 관리 도구인 SCMS, PARMON, GARDMON, ClusterProbe와 비교했을 때, CATS-i는 다양한 자원들을 모니터링하고 관리할 수 있으므로 범용 클러스터 모니터 및 관리 도구로서 적합하다. 특히 SCMS, PARMON, GARDMON, ClusterProbe는 자원의 모니터 및 관리에 중점을 둔 도구이지만, CATS-i는 운영체제 및 프로그램 설치 기능, 배치 작업 제출 및 관리 기능 등을 통합하여 지원하는 특성을 가진다.

5. 결론 및 향후 연구 방향

CATS-i는 물리적 또는 논리적인 다양한 리소스의 모니터 및 관리의 기능, 운영체제와 프로그램 설치 기능, 배치 작업 제출 및 관리 기능 등이 통합되어 있는 강력한 범용 클러스터 관리 도구이다. 또한 안전성이 있고 관리의 신속함을 제공한다. 사용자에게 친숙한 사용자 인터페이스는 클러스터의 상태를 쉽게 파악하고, 편리하게 관리할 수 있다. 그러나 별도의 서버 시스템들이 통합된 도구이므로 시스템의 최적화와 함께, 수백대 이상의 많은 노드들의 효과적인 데이터 관리를 위해 데이터베이스의 필요성이 요구된다. 따라서 향후 이러한 문제점을 해결하기 위한 연구를 진행할 예정이다.

참고 문헌

- [1] Gregory F. Pfister, *In Search of Clusters*, 2nd Edition, Prentice Hall, 1998.
- [2] Rajhumar Buyya, *High Performance Cluster Computing*, Vol 1, Prentice Hall, 1999.
- [3] M. A. Baker, G. C. Fox, and H. W. Yau, "Cluster Computing Review," NPAC Technical Report SCCS-748, 1995.
- [4] D. Ridge, D. Becker, P. Merkey and T. Sterling, "Beowulf: Harnessing the Power of Parallelism in a Pile-of-PCs," Proceedings of the IEEE Aerospace Conference, 1997.
- [5] Z. Liang, Y. Sun, and C.-L. Wang, "ClusterProbe: An Open, Flexible and Scalable Cluster Monitoring Tool," Proceedings of the First International Workshop on Cluster Computing, pp. 261-268, 1999.
- [6] R. J. Allan, S. J. Andrews, M. F. Guest, P. M. Oliver, D. Henty, L. Smith, S. Telford, and S. Booth, "Designing and Building Beowulf-class Cluster Computers," Technical Report UKHEC 2000, 2000.
- [7] Wild Packets Inc., "Applying EtherPeek to Switched Network Management," White Paper, 2001.
- [8] T. Shaughnessy, T. J. Velte, *CISCO A Beginner's Guide*, McGraw-Hill, 2000.
- [9] Tobias Oetiker and Dave Rand, "MRTG: Multi Router Traffic Grapher," <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>
- [10] Sun, *Java Secure Socket Extension(JSSE)*, <http://java.sun.com/products/jsse>.
- [11] A. Bayucan, R. L. Henderson, C. Lesiak, B. Mann, T. Proett, D. Tweten, "Portable Batch System External Reference Specification Release 2.2," MRJ Technology Solutions, 1999.
- [12] 문선영, *클러스터에서의 다중 배치 작업 관리 시스템을 위한 미들웨어*, 석사학위 논문, 숭실대학교, 2000.

- [13] Putchong Uthayopas, Jullawadee Maneesilp and Paricha Ingongnam, "SCMS: An Integrated Cluster Management Tool for Beowulf Cluster System," Proceedings of the International Conference on Parallel and Distributed Proceeding Techniques and Applications 2000 (PDPTA 2000), Las Vegas, Nevada, USA, 2000.
- [14] R. Buyya, K. Mohan and B. Gopal, "PARMON: A Comprehensive Cluster Monitoring System," The Australian Users Group for UNIX and Open Systems Conference and Exhibition, AUUG 98 - Open Systems: The Common Thread, 1998.
- [15] R. Buyya, B. T. Koshy, and R. Mudlapur, "GARDMON: A Java-based monitoring Tool for Gardens Non-dedicated Cluster Computing System," Proceedings of the Sixth International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 99), Las Vegas, Nevada, USA, 1999.

권 성 주

1997년 2월 숭실대학교 소프트웨어 공학과 학사. 2000년 8월 숭실대학교 컴퓨터학과 석사. 2001년 3월 ~ 현재 숭실대학교 컴퓨터학과 박사과정. 관심분야는 모니터링, 컴퓨터 네트워크, 병렬처리

최 재 영

1984년 서울대학교 제어계측공학과 학사
1986년 미국 남가주대학교 컴퓨터공학 석사. 1991년 미국 코넬대학교 컴퓨터공학 박사. 1992년 1월 ~ 1994년 2월 미국 국립 오크리지연구소 연구원. 1994년 3월 ~ 1995년 2월 미국 테네시 주립대학교 연구교수. 2001년 8월 ~ 2002년 8월 미국 국립 슈퍼컴퓨팅 응용센터(NCSA) 초빙연구원. 1995년 3월 ~ 현재 숭실대학교 정보과학대학 컴퓨터학부 조교수/부교수. 관심분야는 초고속 컴퓨팅, 병렬/분산처리 시스템 소프트웨어

김 은 희

1993년 2월 숭실대학교 전자계산학과 학사. 1998년 8월 숭실대학교 대학원 컴퓨터학과 석사. 1998년 9월 ~ 현재 숭실대학교 대학원 컴퓨터학과 박사과정. 관심분야는 병렬/분산 처리, 클러스터 컴퓨팅, 그리드

김 지 연

2000년 2월 숭실대학교 컴퓨터학부 학사
2002년 2월 숭실대학교 대학원 컴퓨터학과 석사. 2002년 2월 ~ 현재 한국전자통신연구원 컴퓨터소프트웨어연구소 연구원. 관심분야는 리눅스 커널, 미디어 스트리밍, 이미지 프로세싱

박 용 관

2000년 2월 숭실대학교 컴퓨터학부 학사
2002년 2월 숭실대학교 대학원 컴퓨터학과 석사. 2002년 3월 ~ 현재 숭실대학교 대학원 컴퓨터학과 박사과정. 관심분야는 컴파일러, 프로그래밍 언어 병렬/분산처리, XML