단일 시스템 이미지를 제공하는 클러스터 유지 관리 소프트웨어의 설계 및 구현

이대우(Dae-Woo Lee) 최민(Min Choi) 박동근(Dong-Geun Park)

한국과학기술원 전자전산학과 전산학전공 컴퓨터구조연구실

2003년 7월 22일

요약

본 보고서에서는 단일 시스템 이미지를 제공하는 클러스터 유지 관리 소프트웨어인 KCMS(KAIST Cluster Management Software)에 관해서 설명한다. KCMS의 특징은 클러스터 상태 확인(Cluster Monitoring)과 작업 관리(Job Management)를 웹 기반의 인터페이스를 통해 단일 시스템 이미지(Single System Image)를 제공한다는 것이다. KCMS는 100Mbps Fast Ethernet으로 연결된 여러 대의 PC로 구성된 클러스터 시스템에서 구현되었으며, 네트워크 파일 시스템(Network File System) 기반에서 동작한다. KCMS에서는 클러스터 상태 확인 기능을 위해 오픈 소스 프로젝트인 ganglia 2.5.3을 이용하였고, 작업 관리 기능을 위해 본 소프트웨어와 병행하게 구현된 부하 분산 시스템(Load Balancing System)을 이용하였다.

제 1 장

서론

본 보고서에서는 단일 시스템 이미지를 제공하는 클러스터 유지 관리 소프트웨어인 KCMS (KAIST Cluster Management Software)에 관해서 설명한다.

KCMS는 여러 대의 PC를 100Mbps Fast Ethernet으로 연결된 클러스터 시스템 상에서 실행된다. 각각의 PC들은 네트워크 파일 시스템(Network File System)을 기반으로 하고, 운영체계로 Linux 2.4.2를 사용한다.

KCMS는 크게 클러스터 상태 확인 모듈(Cluster Monitor Module)과 작업 관리 모듈(Job Management Module), 그리고 인터페이스 모듈로 나뉜다. 클러스터 상태 확인 모듈은 정해진 여러 가지 기준에 따라 KCMS에 등록된 클러스터 및 클러스터에 포함된 노드들의 상태를 사용자에게 보여준다. 이는 오픈 소스 프로젝트인 ganglia 2.5.3을 사용했다. 작업 관리 모듈은 인증된 사용자가 KCMS에 등록된 클러스터 중에서 선택한 하나의 클러스터에 대해 작업을 수행하도록 요청하고 그 결과를 확인할 수 있도록 해준다. 이는 본 소프트웨어와 병행하게 구현된 부하 분산 시스템을 이용하였다. 그리고 인터페이스 모듈은 ganglia webfrontend를 수정하여 단일 시스템 이미지를 제공하도록 하였다.

본 보고서의 구성은 다음과 같다. 우선 2장에서는 KCMS의 설치 및 설정에 관한 내용을 설명하고, 3장에서는 KCMS의 웹 기반 인터페이스를 사용하는 방법에 대해서 설명한다. 4, 5장에서 각각 클러스터 상태 확인 모듈과 작업 관리 모듈에 대해서 살펴본 후, 마지막으로 6장에서 결론을 맺는다.

제 2 장

설치 및 설정

이 장은 KCMS를 사용하기 위한 설치 및 설정 방법에 대해 설명한다.

1. 설치

KCMS는 Linux 2.4.2에서 개발되고 테스트되었으나, 다른 버전의 Linux 환경에서도 잘 동작할 것이다.

설치는 다음과 같이 진행된다. 우선 tar로 묶여있는 파일을 풀게 되면 kcms라는 디렉토리가 생성된다. 이 디렉토리에는 클러스터 상태 확인 모듈의 소스 파일들이 들어있는 cluster_monitor/ 디렉토리와 작업 관리 모듈의 소스 파일들이 들어있는 job_management/ 디렉토리, 그리고 인터페이스 모듈의 소스 파일들이 들어있는 kcms_webfrontend/ 디렉토리로 구성된다. 이제 각각의 모듈들을 설치하기만 하면 된다.

각각의 모듈들을 설치하는 방법은 다음에서 설명한다.

1.1. 클러스터 상태 확인 모듈의 설치

cluster_monitor/ 디렉토리는 ganglia 2.5.3의 소스 파일들이 들어있는 ganglia-monitor-core2.5.3/ 디렉토리와 그래프를 그리는데 사용하는 RRDTool 1.0.43의 소스 파일이 들어있는 rrdtool1.4.3/ 디렉토리, 그리고 이들을 설치하는 방법이 작성되어 있는 INSTALL 파일로 구성된다.

클러스터 상태 확인 모듈은 ganglia 2.5.3의 데몬 프로그램 중에서 두 가지를 사용하며, 이것들은 gmond(Ganglia Monitoring Daemon)와 gmetad(Ganglia Meta Daemon)이다. gmond는 자기가 실행되고 있는 노드의 상태 정보를 수집하고, 같은 클러스터 내에 있는 모든 노드들의 상태 정보를 멀티캐스트(multicast) 통신을 통해서 서로 공유한다. 그리고 gmetad는 gmond들이 공유하는 노드들의 상태 정보들을 수집해서 인터페이스 모듈에게 알려주는 역할을 한다. 따라서 gmond는 KCMS가 관리하려고 하는 모든 노드들에 설치해야 하며, gmetad는 하나의 노드에만 설치하면 된다. 이 때 gmetad는 클러스터들에 포함되지 않는 PC에 설치해도 상관없다. 단, gmond가 설치되는 노드들과 TCP로 통신이 가능한 PC인 경우에만 가능하다.

gmond 및 gmetad의 설치는 다음과 같다.

1. 설정 스크립트를 실행한다.

```
[root@can] sh ./configure
```

이 때 gmetad를 설치하는 경우라면 '--with-gmetad' 라는 옵션을 주어야 하며, 그 전에 RRDTool을 설치해야 한다.

2. 소스 파일을 컴파일한다.

[root@can] make

3. 인스톨한다.

[root@can] make install

이후의 설치 과정은 gmond를 설치하는 경우와 gmetad를 설치하는 경우가 다르기 때문에 나누어서 설명한다.

gmond를 설치하는 경우는 다음과 같다.

4. gmond의 설정 파일(gmond.conf)를 /etc/ 디렉토리에 복사한다.

[root@can] cp ./gmond/gmond.conf /etc/

5. 노드가 새로 부팅되는 경우에 gmond가 자동으로 실행될 수 있도록 해준다.

[root@can] cp ./gmond/gmond.init /etc/rc.d/init.d/ gmond

[root@can] chkconfig --add gmond

6. 설치된 gmond를 실행한다.

[root@can] /etc/rc.d/init.d/gmond start

Starting GANGLIA gmond: [OK]

7. 실행 중인 gmond를 종료하기 위해서는 'start' 대신 'stop'을 옵션으로 준다.

```
[root@can] /etc/rc.d/init.d/gmond stop
Stopping GANGLIA gmond:                    [  OK  ]
```

gmetad를 설치하는 경우는 다음과 같다.

4. gmetad는 gmond들에게서 받아온 정보를 그래프로 나타내기 위해서 파일로 저장한다. 이를 위한 디렉토리를 만들어준다.

```
[root@can] mkdir -p /var/lib/ganglia/rrds
[root@can] chown -R nobody /var/lib/ganglia/rrds
```

5. gmetad의 설정 파일(gmetad.conf)를 /etc/ 디렉토리에 복사한다.

```
[root@can] cp ./gmetad/gmetad.conf /etc/
```

6. 노드가 새로 부팅되는 경우에 gmond가 자동으로 실행될 수 있도록 해준다.

```
[root@can] cp ./gmetad/gmetad.init /etc/rc.d/init.d/gmetad
[root@can] chkconfig --add gmetad
```

7. 설치된 gmetad를 실행한다.

```
[root@can] /etc/rc.d/init.d/gmetad start
Starting GANGLIA gmetad:                   [  OK  ]
```

8. 실행 중인 gmetad를 종료하기 위해서는 ‘start’ 대신 ‘stop’을 옵션으로 준다.

```
[root@can] /etc/rc.d/init.d/gmetad stop
Stopping GANGLIA gmetad:                    [   OK   ]
```

gmetad에서 수집한 정보를 웹에서 그래프로 나타내기 위해서는 RRDTool을 사용한다. RRDTool의 설치는 다음과 같다.

1. 설정 스크립트를 실행한다.

```
[root@can] sh ./configure
```

RRDTool 1.0.43은 기본적으로 /usr/local/rrdtool-1.0.43에 설치된다. 다른 디렉토리에 설치하고 싶은 경우에는 ‘--prefix=(새로운 디렉토리 경로)’라는 옵션을 준다.

2. 소스 파일을 컴파일한다.

```
[root@can] make
```

3. 인스톨한다.

```
[root@can] make install
```

1.2. 작업 관리 모듈의 설치

job_management/ 디렉토리에는 중앙 관리자(Central Manager)의 소스 파일이 들어있는 Cent_manager/ 디렉토리와 스케줄러의 소스 파일이 들어있는 Sched/ 디렉토리, 노드 하나의 상태를 확인하는 모니터(Monitor)의 소스 파일이 들어있는 Monitor/ 디렉토리, 그리고 서버의 도메인 이름을 저장하는 serv_host 파일로 구성된다.

작업 관리 모듈의 설치는 쉽게 끝낼 수 있다. Cent_manager/ 디렉토리와 Monitor/ 디렉토리에서 make를 이용하여 컴파일하면 Cent_manager/ 디렉토리에는 Cent, Monitor/ 디렉토리에는 Monitor라는 실행 파일이 각각 생성되며 이들을 실행하는 것으로 설치를 끝낼 수 있다.

1.3. 인터페이스 모듈의 설치

인터페이스 모듈은 HTML 및 PHP 4.0으로 작성되었기 때문에 특별한 설치 과정은 필요하지 않고 웹 서버가 접근할 수 있는 위치에 kcms_webfrontend/ 디렉토리를 복사하는 것으로 설치를 끝낼 수 있다.

2. 설정

다음에서는 클러스터 상태 확인 모듈과 작업 관리 모듈 및 인터페이스 모듈이 각각 제대로 동작하기 위해 어떻게 설정해야 하는지를 설명한다.

2.1. 클러스터 상태 확인 모듈의 설정

클러스터 상태 확인 모듈이 제대로 동작하기 위해서는 1장에서 설명한 설정 파일인 gmond.conf와 gmetad.conf를 정확히 작성해야만 한다. 2.1.1에서는 하나의 클러스터에 대해서 클러스터 상태 확인 모듈을 동작하도록 하기 위한 gmond.conf 설정 방법을 설명하며, 2.1.2에서는 여러 개의 클러스터를 등록하기 위한 gmetad.conf의 설정 방법을 설명한다.

2.1.1. gmond.conf 작성 방법

gmond.conf를 열어보면 설정을 위한 많은 옵션들이 작성되어 있다. 이들 모두를 수정해야할 필요는 없으며 다음과 같이 5개의 옵션만 수정하면 된다. 5개 이외의 옵션에 대한 설정은 gmond.conf에 적혀있는 주석들을 읽어보길 바란다.

- name

  현재 노드가 포함되는 클러스터의 이름을 뜻하며 기본값은 unspecified이다.

- mcast_channel, mcast_port

  mcast_channel 및 mcast_port는 클러스터에 설치된 gmond들의 멀티캐스트 통신을 위한 IP 주소 및 포트 번호를 뜻하며, 같은 클러스터에 설치된 gmond들의 설정에서는 여기에 같은 값을 적어야 한다. 기본값은 239.2.11.71과 8649이다.

- xml_port

  xml_port는 gmetad가 이 gmond에게서 노드의 상태 정보를 얻어올 때 사용할 포트 번호를 뜻하며 기본값은 8649이다.

- trusted_hosts

  마지막으로 trusted_hosts는 이 gmond가 가지고 있는 클러스터의 상태 정보들을 읽어갈 수 있는 PC의 IP 주소를 뜻하며, 여기에 gmetad가 설치된 PC의 IP 주소를 적어야 한다.

설정이 끝나면 다음과 같은 방법으로 gmond가 제대로 동작하는지 확인한다.

[root@can root] telnet localhost 8649

만약 xml_port에 다른 값을 적었다면 8649 대신 그 값을 써야 한다. gmond가 제대로 동작한다면 gmond가 공유하고 있는 노드들의 상태 정보가 XML(eXtensible Markup Language) 문법에 맞춰서 화면에 출력되는 것을 확인할 수 있다.

2.1.2. gmetad.conf 작성 방법

gmetad.conf를 열어보면 설정을 위한 많은 옵션들이 작성되어 있다. 이들 모두를 수정해야할 필요는 없으며 다음과 같이 3개의 옵션만 수정하면 된다. 3개 이외의 옵션에 대한 설정은 gmetad.conf에 적혀있는 주석들을 읽어보길 바란다.

- data_source

  data_source는 클러스터의 상태 정보를 읽어올 gmond의 위치 정보를 뜻한다. 작성하는 방법은 다음과 같다.

  data_source “cluster1” 15 1.1.1.1:8000 1.1.1.2:8000 …

  data_source “cluster2” 15 1.1.2.1:7000 1.1.2.2:7000 …

  …

  cluster1과 cluster2는 클러스터의 이름으로 gmond.conf의 name에 적은 이름과 동일해야 한다. 그 다음의 15는 gmetad가 gmond들에게서 15초마다 한 번씩 노드들의 상태 정보를 읽어오겠다는 것을 의미한다. 그 다음에 적힌 IP 주소와 포트 번호들은 상태 정보를 읽어올 gmond들의 위치 정보를 의미하며, 이 때 포트 번호는 gmond.conf에서의 xml_port에 적은 포트 번호와 동일해야 한다. 이런 식으로 KCMS가 관리할 클러스터의 개수만큼 이를 적어주면 된다.

  같은 클러스터 내에서 실행되고 있는 gmond들은 상태 정보를 공유하고 있기 때문에 gmetad는 이들 중 어디에서나 같은 정보를 얻어올 수 있다. 그러나 위의 예제처럼 하나가 아닌 여러 개의 위치 정보를 적는 이유는 만약 gmetad가 원래 정보를 얻어오고 있던 gmond와의 통신이 중단되었을 때 (예를 들면, 노드가 다운되었을 때) 다른 노드의 gmond로부터 정보를 얻어올 수 있도록 하기 위해서이다.

- gridname

  gridname은 그리드(grid)의 이름을 뜻하며, 여기서 그리드란 말은 단지 등록된 클러스터의 집합을 칭한다.

- xml_port

  마지막으로 xml_port는 인터페이스 모듈이 gmetad로부터 정보를 읽어올 때 사용하는 포트 번호이다.

설정이 끝나면 다음과 같은 방법으로 gmetad가 제대로 동작하는지 확인한다.

[root@can root] telnet localhost 8649

만약 xml_port에 다른 값을 적었다면 8649 대신 그 값을 써야 한다. gmetad가 제대로 동작한다면 gmetad가 얻어오고 있는 모든 노드들의 상태 정보가 XML 문법에 맞춰서 화면에 출력되는 것을 확인할 수 있다.

2.2. 작업 관리 모듈

작업 관리 모듈의 설정은 간단하다. 작업 관리 모듈은 serv_host 파일에 중앙 관리자가 수행되고 있는 노드의 도메인 이름을 적는 것으로 설정을 끝낼 수 있다. 단, 주의해야 할 점은 serv_host 파일이 작업 관리 모듈의 실행 파일들이 수행되고 있는 상위 디렉토리에 존재해야 한다는 점이다.

2.3. 인터페이스 모듈

KCMS의 인터페이스 모듈을 PHP와 연동이 가능한 웹 서버가 설치되었다면 바로 수행할 수 있다. 하지만 작업 관리 모듈을 사용하기 위한 사용자 인증 과정으로 아파치(Apache) 웹 서버에서 지원해주는 방법을 사용하였기 때문에 아파치 웹 서버를 사용하는 것을 권장한다. 다른 웹 서버를 사용하면 사용자 인증이 불가능하기 때문에 작업 관리 모듈을 누구나 사용할 수 있게 된다는 문제점이 생긴다.

인터페이스 모듈은 아파치 웹 서버가 사용자 인증을 지원하도록 설정하는 것과 conf.php를 작성하는 것으로 설정을 완료할 수 있다. 사용자 인증 과정은 아파치 웹 서버의 기본 인증(Basic Authentication)을 이용하여 kcms_webfrontend/member/ 디렉토리의 접근 제한을 설정하는 것으로 가능하게 하였다. 만약 사용자의 이름으로 root를 사용하면 이는 전체 관리자 권한을 갖기 때문에 KCMS의 모든 사용자가 하는 작업을 알 수 있다. 아파치 웹 서버에서의 사용자 인증의 구체적인 내용은 http://apache.kr.net/documents/user_auth.html를 참고한다. conf.php는 파일 이름에서 알 수 있듯이 PHP 파일이므로 이를 수정하는 것은 PHP 소스를 수정한다는 것을 의미하며, PHP 문법에 어긋나지 않도록 수정해야 한다.

conf.php에서 수정해야 하는 부분들은 다음과 같다.

- $gmetad_root, $rrds

  $gmetad_root 및 $rrds는 gmetad가 수집한 정보들을 그래프로 나타내기 위해 저장하는 파일의 디렉토리를 뜻하며 기본적으로 앞의 gmetad 설치 과정에서 /var/lib/ganglia/rrds로 설정한 것과 동일하게 작성되어 있다.

- RRDTOOL

  RRDTOOL은 RRDTool의 실행 파일의 경로를 의미하며 (RRDTool이 설치된 디렉토리)/bin/rrdtool이라고 작성하면 된다.

- $ganglia_ip, $ganglia_port

  $ganglia_ip와 $ganglia_port는 gmetad에서 정보를 읽어오기 위한 IP 주소와 포트 번호이며 $ganglia_ip에는 gmetad가 설치된 노드의 IP 주소, $ganglia_ port는 gmetad.conf의 xml_port 값을 쓰면 된다.

- $default_range

  $default_range는 그래프에 나타나는 기본 범위를 의미하며 hour, day, week, month, year의 다섯 가지 중에서 하나를 고르면 된다.

- $cm_address, $cm_port

  $cm_address와 $cm_port는 작업 관리 모듈의 중앙 관리자들의 IP 주소와 포트 번호를 의미하며, =>에 대해 왼쪽에는 gmond.conf 및 gmetad.conf의 클러스터 이름을, 그리고 오른쪽에는 해당 클러스터를 관리하는 중앙 관리자가 설치된 노드의 IP 주소와 포트 번호를 쓰면 된다.

- $submit_result_path

마지막으로 $submit_result_path는 사용자가 할당하는 작업의 수행 결과가 저장된 파일들이 저장되는 디렉토리의 위치를 뜻하며, 네트워크 파일 시스템에 존재하는 디렉토리로 작성해야만 올바른 동작이 가능하게 된다.

제 3 장

웹 기반 인터페이스

이 장은 KCMS를 설치한 후, 웹 기반 인터페이스를 이용한 KCMS를 사용 방법에 대해서 설명한다. KCMS의 사용은 크게 사용자 인증과 상관없는 클러스터 상태 확인 과정과 사용자 인증 후에만 가능한 작업 관리 과정으로 나뉜다.

1. 클러스터 상태 확인 과정

클러스터 상태 확인 과정은 클러스터 상태 확인 모듈이 지원해주는 기능으로 사용자 인증과 상관없기 때문에 누구나 가능하다.

그림 1은 인터넷 익스플로러 6.0을 이용하여 KCMS에 접속한 화면이다. CAN 그리드라는 이름으로 can1_4, can49_52, can53_55라는 세 개의 클러스터가 KCMS에 등록되어 있다는 것을 보여주고 있다. 초기 화면에서는 등록된 클러스터들의 전체적인 상태를 대략적으로만 보여주고 있으며, 특정한 하나의 클러스터나 하나의 노드의 상태를 보고 싶다면 이들을 보여주는 화면으로 넘어가야 한다. 특정한 하나의 클러스터의 상태를 보고 싶다면 클러스터 이름을 선택하거나 아니면 CAN Grid 옆의 풀다운(Pulldown) 메뉴에서 선택하면 되고, 특정한 하나의 노드의 상태를 보고 싶다면 원하는 노드를 나타내는 그림을 선택하면 된다.

가장 위의 메뉴에는 Monitor와 Login이 있다. Monitor 메뉴를 선택하면 어느 화면에서던지 다시 지금의 초기 화면으로 돌아오게 된다. Login 메뉴는 사용자 인증을 위한 것이기 때문에 나중에 설명한다.

그 아래 메뉴에는 새로 고침 버튼의 역할을 하는 Refresh 버튼이 있고, 그 옆에 그래프의 가시 범위를 나타내는 Last 메뉴가 있다. 현재는 지난 하루 동안의 상태를 그래프로 보여주고 있다는 것을 의미한다. 그리고 아래의 두 개의 그래프는 현재 KCMS에 등록된 세 개의 클러스터들에 포함된 모든 노드들의 동작 여부 및 걸리는 부하, 그리고 메모리 사용률을 나타낸다.

그래프 아래에는 각각의 클러스터에 포함되는 노드들의 상태를 간략하게 볼 수 있다. 각각의 노드는 컴퓨터처럼 생긴 그림으로 나타나며 그림의 색으로 해당하는 노드에 걸리는 부하를 확인할 수 있다. 색깔의 의미와 그 옆의 수치들이 나타내는 값의 의미는 클러스터 이름 옆에 Legend라는 링크를 선택하면 이들을 설명해주는 화면이 나타난다.

그림 2는 can49_52 클러스터를 선택했을 때, can49_52 클러스터의 보다 자세한 상태를 보여주는 화면이다. 그림 1과 비슷한 내용을 보여주고 있지만 범위가 can49_52 클러스터로 한정된 것을 확인할 수 있다. 평균 CPU 사용률의 그래프가 추가되었다.

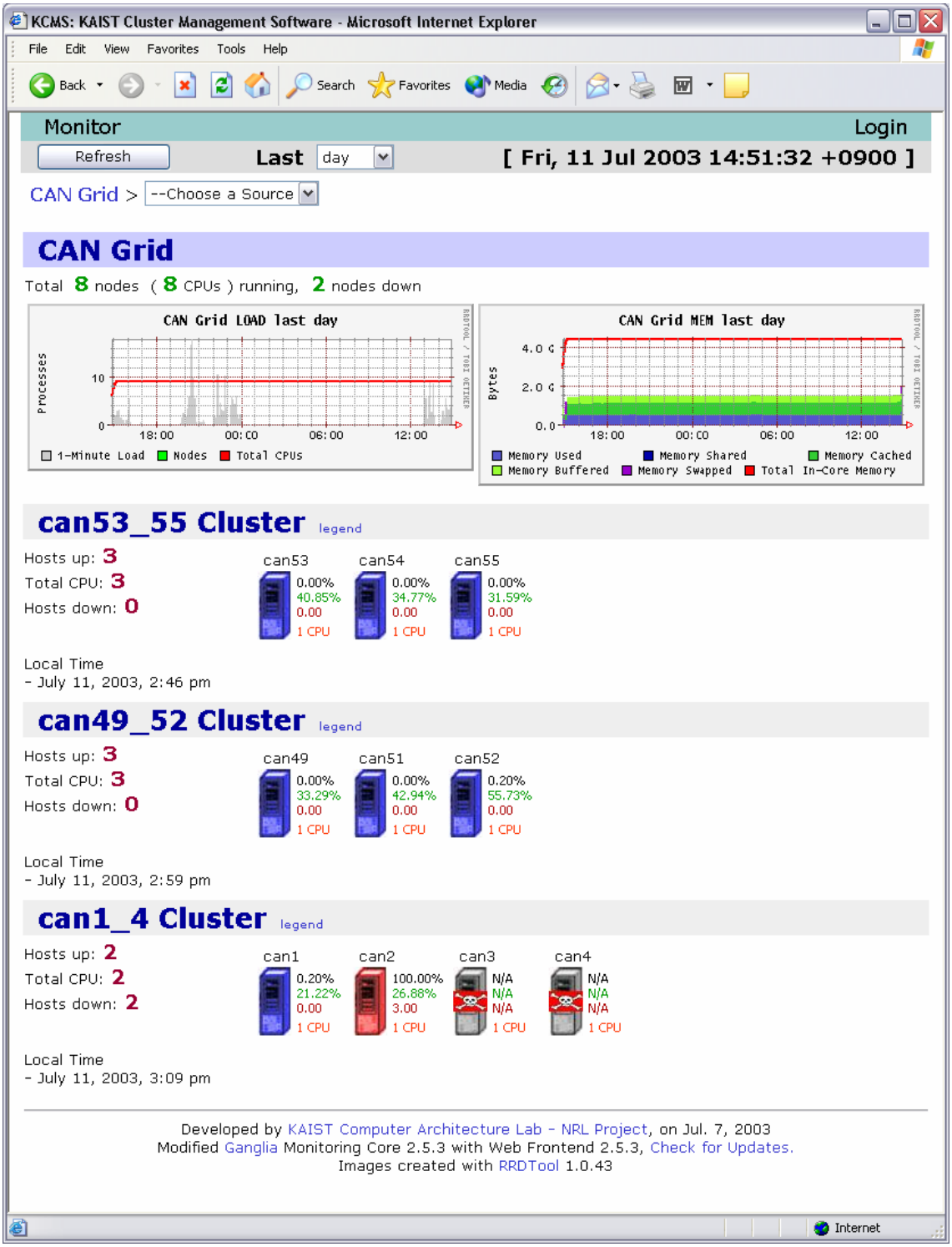

그림 1. KCMS의 초기 화면

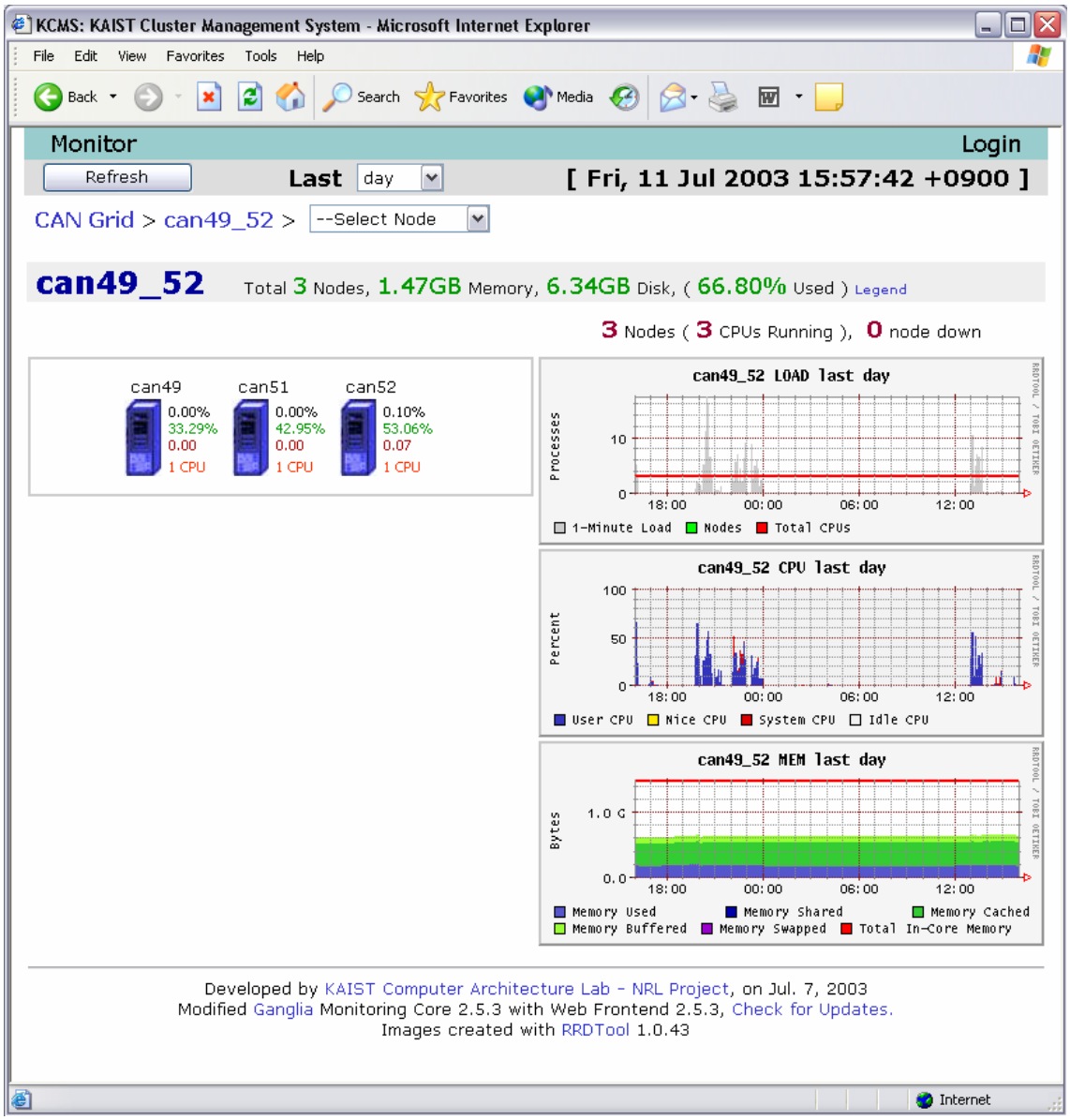

그림 2. can49_52 클러스터의 상태 화면

그림 3-1과 그림 3-2는 can51.kaist.ac.kr 노드에 대한 상태 화면이다. 모든 그래프의 범위는 하나의 노드로 한정되었다. 그림 1, 2와는 달리 여러 가지 기준에 따른 그래프를 보여주고 있다. 원래 ganglia에서는 보다 많은 기준들에 대한 노드들의 상태를 보여주고 있지만, KCMS에서는 중요하다고 생각하는 것들에 대한 상태만 제공하고 있다. 그 기준들은 다음과 같다.

- cpu_idle (%)

  사용되고 있지 않는 CPU의 비율

- cpu_system (%)

  OS의 CPU 점유율

- cpu_user (%)

  사용자 레벨의 프로세스들의 CPU 점유율

- load_fifteen

  15분 동안의 평균 부하값

- load_five

  5분 동안의 평균 부하값

- load_one

  1분 동안의 평균 부하값

- mem_buffers (KB)

  버퍼로 사용되고 있는 메모리의 양

- mem_free (KB)

  사용되고 있지 않는 메모리의 양

- pkts_in (개/sec)

  노드로 들어오고 있는 패킷의 양

- pkts_out (개/sec)

  노드로부터 나가고 있는 패킷의 양

- proc_total (개)

  실행되고 있는 모든 프로세스들의 수

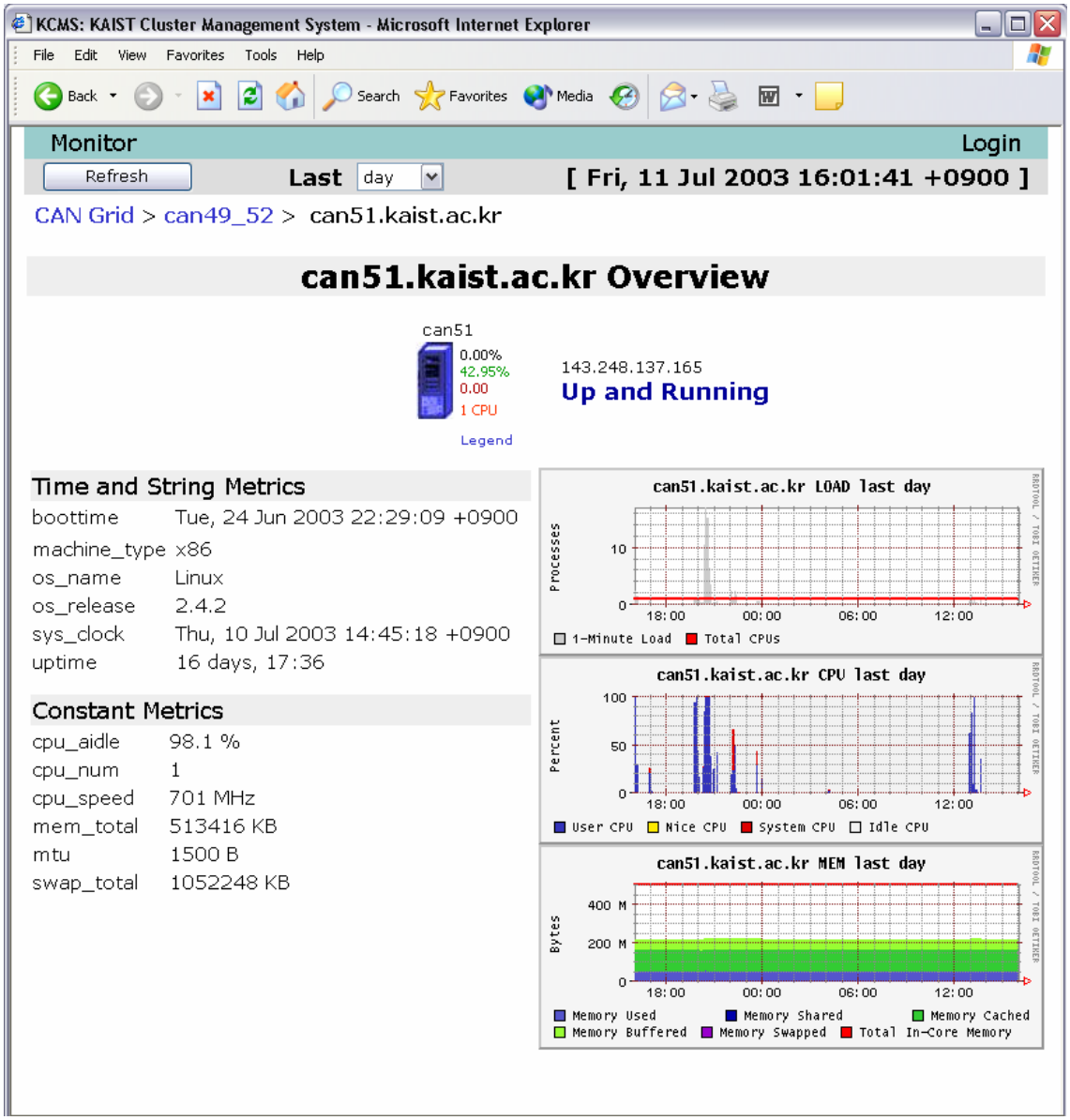

그림 3-1. can51.kaist.ac.kr의 상태 화면

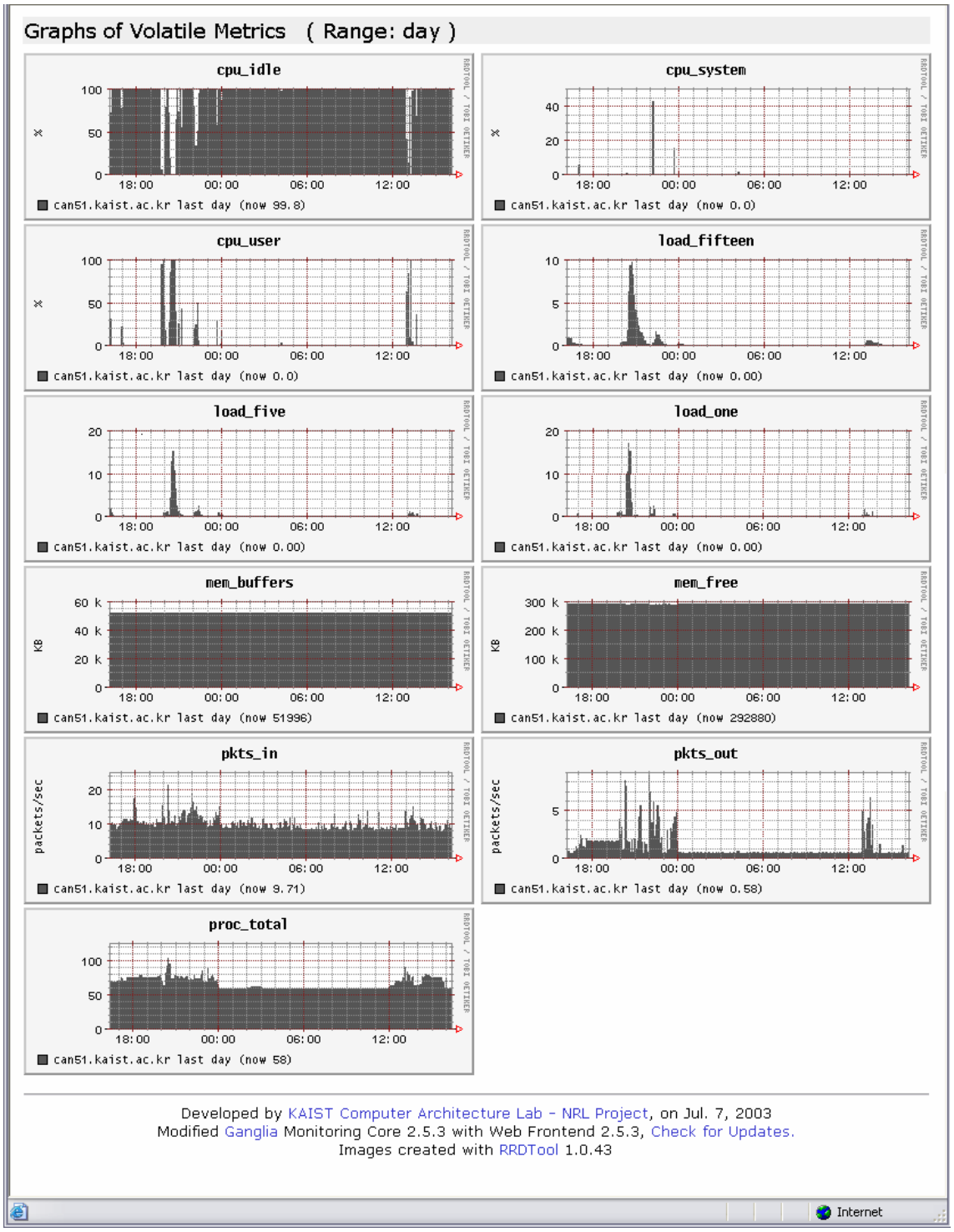

그림 3-2. 여러 가지 기준에 따른 can51.kaist.ac.kr의 상태 그래프

2. 작업 관리 과정

작업 관리 과정은 작업 관리 모듈이 지원해주는 기능으로 악의적인 사용을 막기 위하여 인증된 사용자만이 가능한 기능이다.

사용자 인증을 받기 위해서는 오른쪽 상단의 Login 메뉴를 선택하면 된다. 그러면 그림 4와 같은 인증창이 뜨게 되는데, 이 때 등록된 사용자명과 패스워드를 입력하면 된다. (2장의 2.3절에서 설명) 그림 4는 Windows XP에서의 인증창 모습이며 이는 운영체계마다 다르다.

그림 4. 사용자 인증

사용자 인증 후에 로그아웃하는 기능은 없으며, 그렇기 때문에 인증 후에 사용을 마친 다음에는 항상 브라우저를 종료하는 것이 안전하다.

사용자 인증을 받게 되면 다시 초기화면으로 돌아가게 된다. 클러스터 상태 확인은 여전히 가능하기 때문에 겉보기엔 인증 전과 별 차이가 없지만 없어지거나 새로 나타난 메뉴가 있다. 오른쪽 상단의 Login 메뉴는 없어지며, 특정 클러스터를 선택하면 인증 전에는 없었던 Process Monitor, Job Submit, Job Results 버튼들이 나타난다. Process Monitor는 수행되고 있는 프로세스들의 정보를 확인할 수 있는 화면을 보여주며, 확인한 후에 사용자로 하여금 수행되고 있는 프로세스를 다른 노드로 이동(migration)시키거나 종료시킬 수 있게 한다. Job Submit은 사용자가 원하는 작업을 클러스터에 할당하기 위한 정보를 입력하는 화면을 보여주며, Job Results는 Job Submit 메뉴를 이용하여 할당된 작업들의 수행 상태 및 결과를 보여준다.

그림 5는 can1_4 클러스터에 대해 Process Monitor 메뉴를 선택했을 때의 화면이다. 하나의 노드에 대해 각각의 열이 하나의 프로세스에 대한 정보이며 순서대로 PID값, 수행 상태 – R: 수행 중, T: 종료 – , 그리고 이름을 나타낸다.

수행되고 있는 프로세스를 다른 노드로 이동시키거나 종료시키기 위해서는 먼저 원하는 하나의 프로세스를 선택한 후, 각각 Migration, Kill 버튼을 누르면 된다. Migration 버튼을 누르는 경우에는 어느 노드로 옮길지 선택할 수 있는 메뉴가 나타난다.

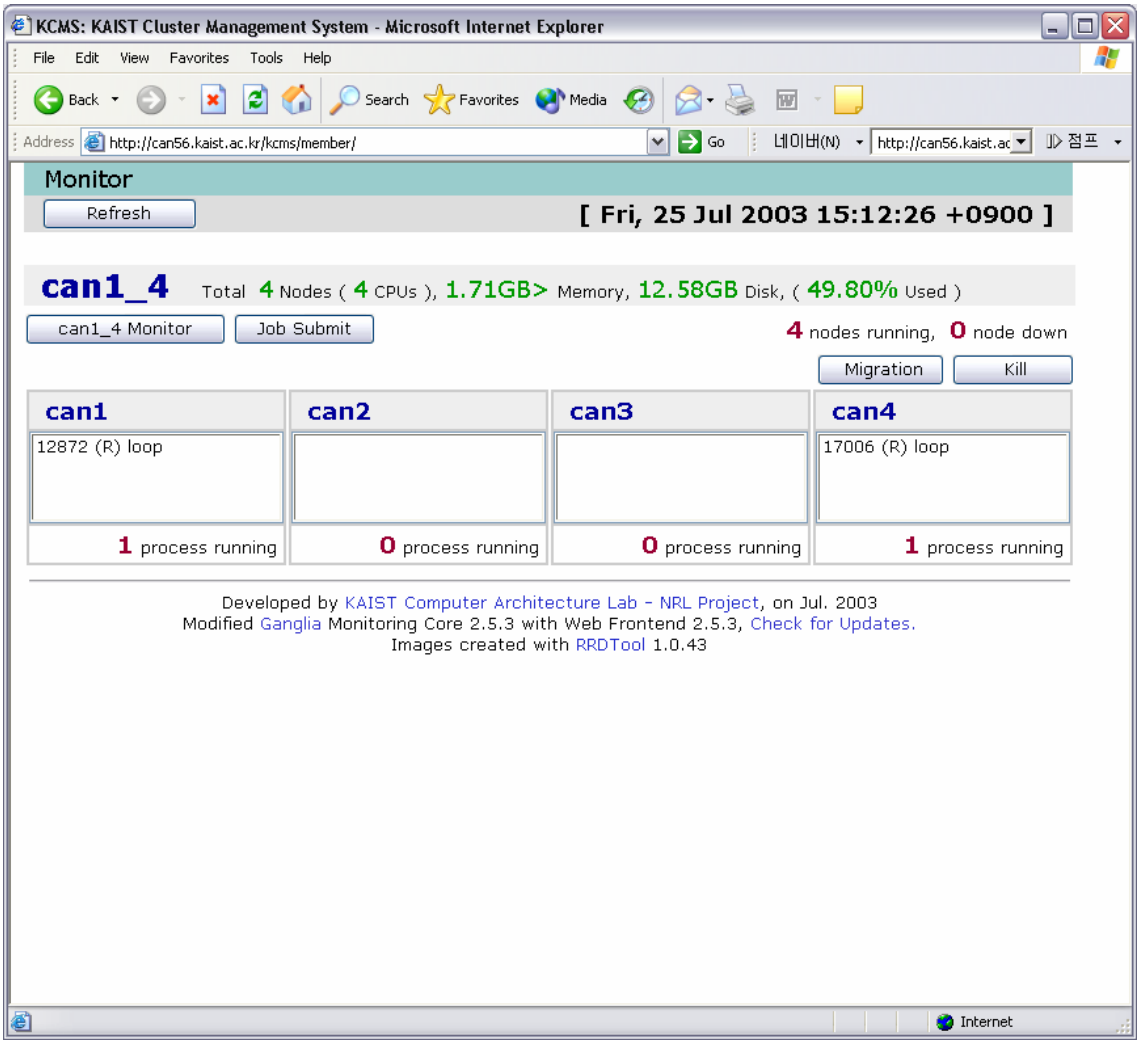

그림 5. 수행되고 있는 프로세스 확인

그림 6은 Job Submit을 선택한 경우이다. 작업을 할당하기 위해서는 다음과 같은 정보를 입력해야 한다.

- Job Name

  각각의 작업들을 구분하기 위한 이름을 의미한다.

- No. of Arguments

  작업을 수행하기 위해서 필요한 인자들의 개수

- Arguments

  작업을 수행하기 위해서 필요한 인자들

- Job Classification

  작업의 종류로 단일 수행 작업을 의미하는 Single, 일괄 처리 작업을 의미하는 Batch, 그리고 병렬 수행 작업을 의미하는 Parallel을 선택할 수 있다.

- Input File

  입력 파일의 이름을 의미한다.

- Output File

  출력 파일의 이름을 의미한다.

- Error Log File

  작업 수행 도중에 오류가 발생하는 경우, 그에 대한 기록을 남기기 위한 로그 파일의 이름을 의미한다.

- Executing File

  작업을 수행시키기 위한 실행 파일 이름을 의미한다. 이 때, 수행하려는 작업 프로그램은 모두 네트워크 파일 시스템 상에 존재해야 한다.

- Priority

  수행하려는 작업의 우선 순위를 의미한다.

- Base Directory

  작업 프로그램이 수행되는 기본 디렉토리를 의미한다.

- Memory Size

  작업 수행을 위해 할당할 메모리의 크기를 의미하며, 단위는

- No. of Tasks / Nodes

  일괄 처리 작업이나 병렬 수행 작업의 경우에는 하나의 작업을 수행하기 위해 여러 개의 프로세스가 수행되어야 한다. 이 때, 각각의 노드에서 수행시킬 프로세스의 개수를 의미한다.

- No. of Total Tasks

  일괄 처리 작업이나 병렬 수행 작업을 위해서 이용할 프로세스의 개수이다.

- RSS Limit

  사용자가 작업에 사용되는 메모리의 최대 사용량을 제한하고 싶은 경우에 그 최대값을 의미하며, 현재 부하 분산 시스템에 적용되지는 않는다.

- Restart from Check Point

  작업이 비정상적으로 종료되었을 경우에 체크포인트 지점에서 자동으로 다시 수행시킬 것인지를 의미한다.

- Load Metric

  노드들의 부하 분산을 위해 각각의 노드들의 부하를 측정하는 기준을 의미한다. Load average는 부하값을, CPU queue length는, Load + Memory는 부하값과 메모리 이용량을 기준으로 삼는다.

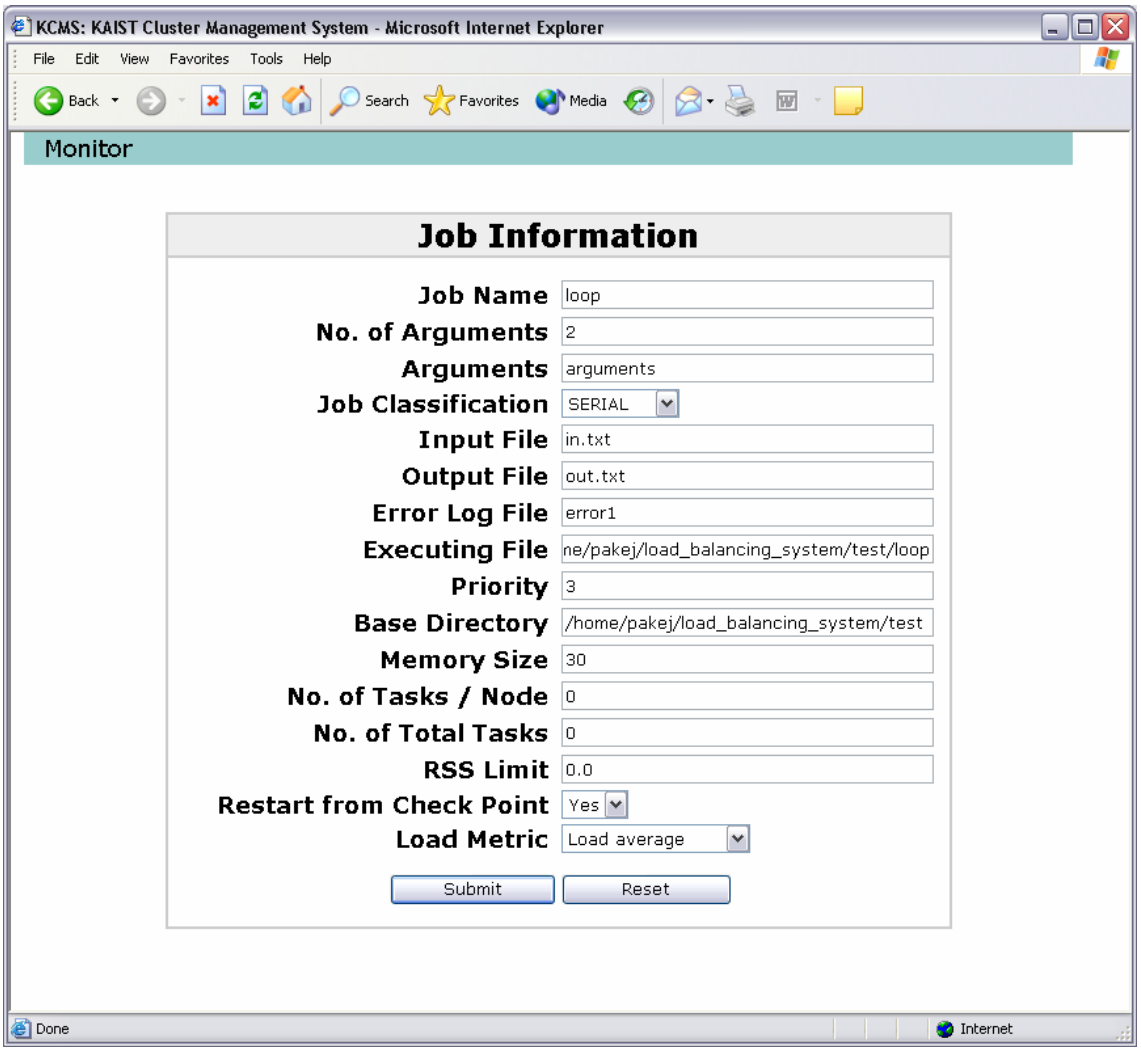

그림 6. 작업 할당을 위한 정보 입력.

그림 7은 Job Results를 선택한 경우이다. 나타나는 작업들의 정보는 지금까지 사용자가 현재 계정을 이용하여 할당한 모든 작업들에 대한 것들이다. 즉, 이전에 KCMS를 이용하여 할당했던 작업들의 정보들도 모두 나타난다. 최근에 할당했던 것들일수록 아래쪽에 존재한다.

3. 프로토콜

KCMS에서는 클러스터 상태 확인 모듈로서 ganglia 2.5.3을 이용하고, 작업 관리 모듈로서 본 소프트웨어와 병행하게 구현된 부하 분산 시스템을 이용한다. 그러나 이 두 모듈 및 인터페이스 모듈은 모두 독립적으로 구현되었기 때문에 그 사이의 통신을 위한 프로토콜만 맞는다면 어떠한 모듈이라도 사용할 수 있다. ganglia와 부하 분산 시스템은 독립적으로 동작하기 때문에 통신을 하지 않기 때문에 제외하고, 이 장에서는 인터페이스 모듈과 클러스터 상태 확인 모듈, 그리고 인터페이스 모듈과 작업 관리 모듈 사이의 프로토콜에 대해 설명한다. 모든 프로토콜은 텍스트 기반으로 TCP를 이용한다.

3.1. 인터페이스 모듈과 클러스터 상태 확인 모듈

클러스터 상태 확인을 위해서는 인터페이스 모듈이 클러스터 상태 확인 모듈로부터 정보를 읽어오는 경우만 존재한다. 이 때 XML 포맷을 이용하여 통신을 하며, 문법은 ganglia에서 사용하고 있는 XML 문법을 그대로 이용하였으며 불필요한 부분을 제외한 나머지를 DTD(Document Type Definition)으로 나타낸 것은 다음과 같다. 몇 가지 속성(attribute)의 설명은 첨부하였다.

```
<!ELEMENT GANGLIA_XML (GRID)*>
    <!ATTLIST GANGLIA_XML VERSION CDATA #REQUIRED>
    <!ATTLIST GANGLIA_XML SOURCE CDATA #REQUIRED>
```

<!ELEMENT GRID (CLUSTER)>*

<!ATTLIST GRID NAME CDATA #REQUIRED>

<!ELEMENT CLUSTER (HOST)>*

<!ATTLIST CLUSTER NAME CDATA #REQUIRED>

<!ATTLIST CLUSTER LOCALTIME CDATA #REQUIRED>

<!ELEMENT HOST (METRIC)>*

<!ATTLIST HOST NAME CDATA #REQUIRED>

<!ATTLIST HOST IP CDATA #REQUIRED>

<!ATTLIST HOST TN CDATA #IMPLIED>

<!ATTLIST HOST TMAX CDATA #IMPLIED>

<!ATTLIST HOST GMOND_STARTED CDATA #IMPLIED>

<!ELEMENT METRIC EMPTY>

<!ATTLIST METRIC NAME CDATA #REQUIRED>

<!ATTLIST METRIC VAL CDATA #REQUIRED>

- GRID/NAME

  그리드의 이름을 의미한다.

- CLUSTER/NAME

  클러스터의 이름을 의미한다.

- CLUSTER/LOCALTIME

  클러스터 내에서의 현재 시간을 의미한다.

- HOST/NAME

  노드의 이름을 의미한다.

- HOST/IP

  노드의 IP 주소를 의미한다.

- HOST/TN

  이 노드에서 수행하고 있는 gmond가 정보를 보내줬을 때부터 지금까지 흐른 시간으로 단위는 초이다.

- HOST/TMAX

  TN의 한계값으로 만약 TN이 TMAX보다 크면 노드가 동작하지 못하고 있다는 것을 의미한다.

- METRIC/NAME

  노드의 상태를 판단하기 위한 기준의 이름을 의미한다.

- METRIC/VAL

  기준에 대한 측정값을 의미한다.

3.2. 인터페이스 모듈과 작업 관리 모듈

인터페이스 모듈과 작업 관리 모듈 간의 통신은 세 가지 경우가 있다.

- 프로세스 정보를 얻어오는 경우
- 프로세스를 다른 노드로 이동시키는 경우
- 프로세스를 종료시키는 경우

프로세스 정보를 얻어오는 경우에는 우선 인터페이스 모듈에서 사용자 이름을 특정 클러스터에 해당되는 작업 관리 모듈에게 보내주면, 작업 관리 모듈에서는 클러스터에 포함된 모든 노드들에서 수행되고 있는 프로세스들의 이름과 PID,

그리고 상태값을 – 0: 수행 중, 1: 종료 – 인터페이스 모듈에게 보내준다. 다음은 인터페이스 모듈에서의 메시지 규칙이다. (send: 보내는 메시지, receive: 받는 메시지)

send = REQ '|' user

receive = (node '|' (pid '|' pname '|' pstate '&'))**

user = 사용자 이름

pid = 프로세스의 PID

pname = 프로세스의 이름

pstate = 프로세스의 상태

프로세스를 다른 노드로 이동시키는 경우에는 인터페이스 모듈에서 해당하는 프로세스의 PID와 원래 수행되던 노드의 IP 주소나 domain name, 그리고 이동할 노드의 IP 주소나 domain name을 작업 관리 모듈로 보내주면 된다. 다음은 그 메시지 규칙이다.

send = MIG '|' pid '|' from '|' to

pid = 이동할 프로세스의 PID

from = 프로세스가 원래 수행되던 노드의 IP 주소나 domain name

to = 이동할 노드의 IP 주소나 domain name

프로세스를 종료시키는 경우에는 인터페이스 모듈에서 해당하는 프로세스의 PID와 수행되고 있는 노드의 IP 주소나 domain name을 작업 관리 모듈로 보내주면 된다. 다음은 그 메시지 규칙이다.

send = KIL '|' pid '|' node

pid = 종료시킬 프로세스의 PID

node = 프로세스가 수행되고 있는 노드의 IP 주소나 domain name

제 4 장

클러스터 상태 확인 모듈

이 장은 클러스터 상태 확인 모듈에 대해서 설명한다. KCMS의 클러스터 상태 확인 모듈은 오픈 소스 프로젝트인 ganglia 2.5.3을 이용하였으며, 이 장에서는 이 ganglia에 대해서 설명한다.

ganglia는 클러스터 상태 확인을 위해 개발된 소프트웨어로 계층 구조(hierarchical structure)를 기반으로 하고 있고 상위 계층과 통신할 때는 구조화된 정보를 지원하기 용이한 XML 언어를 이용하고 있기 때문에 확장성이 뛰어나다. 여기서 계층 구조를 이루는 것들이 앞에서 언급했던 gmond와 gmetad이다.

gmond는 수행하고 있는 노드의 정보를 수집하는 역할을 하는 프로그램이다. 좀더 자세하게 설명하면 gmond를 구성하는 세 가지 쓰레드(thread)가 다음과 같은 작업을 수행한다.

- 노드의 정보를 수집하여 멀티캐스트 채널에 전달
- 멀티캐스트 채널에서 클러스터 내의 모든 노드들에 대한 정보를 수신하여 저장
- 멀티캐스트 채널에서 받은 정보를 XML 언어로 상위 계층에게 전달

따라서 클러스터 내의 모든 gmond들은 모두 같은 정보를 저장하고 있기 때문에 동작하지 않는 노드가 존재한다고 하더라도 동작하고 있는 노드가 존재하는 한 상위 계층은 클러스터의 현재 상태를 알아낼 수 있다는 장점이 있다.

gmetad는 gmond보다 상위 계층을 구성한다. gmetad는 XML 언어를 읽어 들이기 때문에 gmond가 전해주는 클러스터 정보뿐만 아니라 또 다른 gmetad가 전해주는 정보들도 읽어 들일 수 있다. ganglia 내부적으로는 gmond가 전해주는 정보를 클러스터 정보라고 하고, 또 다른 gmetad가 전해주는 정보를 그리드 정보라고 부른다.

따라서 gmetad끼리도 계층 구조를 구성할 수가 있다. 즉, ganglia의 계층 구조는 gmond가 말단, 나머지는 모두 gmetad로 구성되며, 이 때문에 ganglia는 높은 확장성을 가지고 있다. 단, KCMS에서는 부하 분산 시스템과의 연동을 위해서 gmetad끼리의 계층 구조를 허락하지 않고 gmetad는 하나만 동작하게 한다.

다음은 ganglia의 계층 구조를 그림으로 나타낸 것이다.

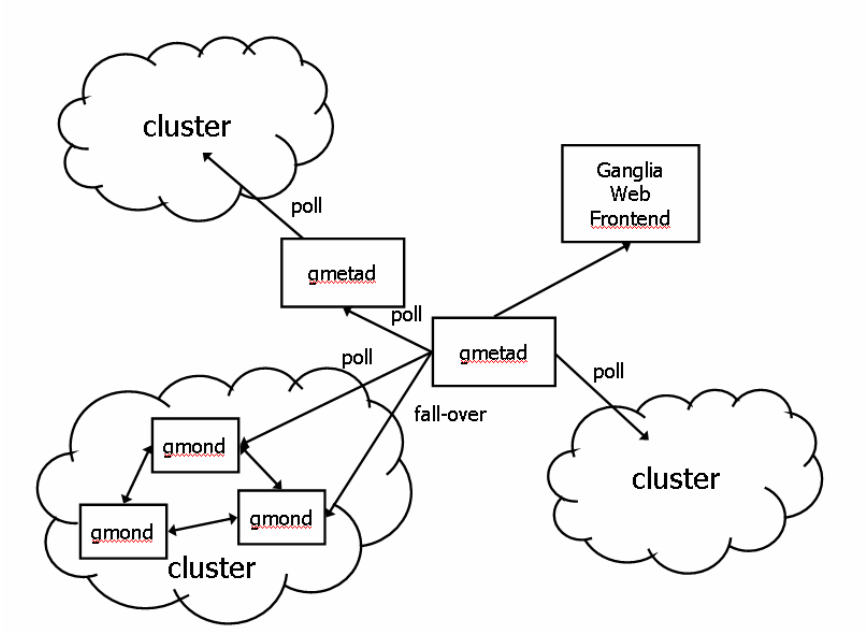

그림. ganglia의 계층 구조

제 5 장

작업 관리 모듈

이 장은 작업 관리 모듈에 대해서 설명한다. KCMS의 작업 관리 모듈은 본 소프트웨어와 병행하게 구현된 부하 분산 시스템을 이용하며, 이 장에서는 이 부하 분산 시스템에 대해 간략히 설명한다.

부하 분산 시스템은 하나의 노드에서 수행되는 중앙 관리자와 클러스터 내의 모든 노드들에서 수행되는 모니터로 구성된다. 이 때, 중앙 관리자는 서버 역할을 하고 나머지 모니터들은 클라이언트 역할을 한다.

중앙 관리자는 모니터들과 통신하면서 클러스터 내의 전체적인 작업 관리를 총괄적으로 담당한다. 즉, 클러스터에서 수행하고 있는 작업들의 정보를 관리하고 클러스터에 작업이 할당되면 각각의 노드들에게 적당히 분배해준다. 각각의 모니터들은 수행되고 있는 노드들의 상태를 중앙 관리자에게 알려주며, 또한 중앙 관리자가 할당하는 작업을 수행하고 작업 정보를 관리한다.

다음 그림은 부하 분산 시스템의 전체적인 구조를 나타낸다.

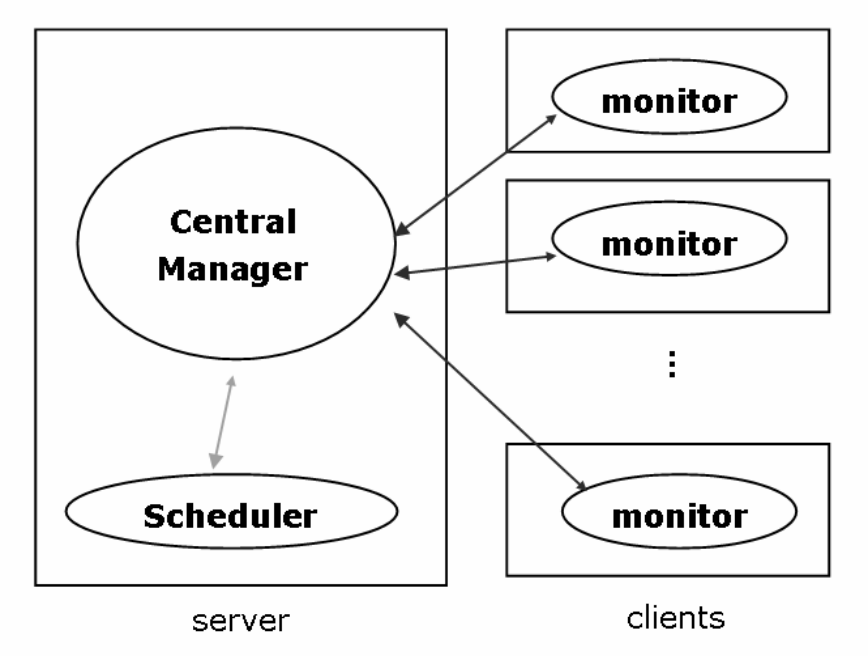

그림. 부하 분산 시스템의 구조

제 6 장

결론

본 보고서에서는 단일 시스템 이미지를 제공하는 클러스터 유지 관리 소프트웨어인 KCMS에 대해서 설명하였다. KCMS의 특징은 클러스터 상태 확인 기능 및 작업 관리 기능을 가지고 있으며, 이러한 작업을 사용하기 편리한 웹 기반 인터페이스를 통해 하나의 PC에서 수행할 수 있기 때문에 단일 시스템 이미지를 제공한다는 것이다. 또한 이 두 가지 기능을 서로 독립적인 모듈들이 제공해주기 때문에 모듈 간의 프로토콜만 유지한다면 모듈의 변경이 용이하다는 점도 하나의 장점으로 생각할 수 있다.

그러나 아직 단일 시스템 이미지를 완전하게 제공하는 것은 아니기 때문에 이를 위해서는 추가적인 개발이 필요할 것이다.

참고 자료

[1] M. Massie, B. Chun, and D. Culler. *The Ganglia Distributed Monitoring System: Design, Implementation, and Experience*. Feb. 2003.

[2] Ganglia Development Homepage. http://ganglia.sourceforge.net

[3] SCMS Cluster Management Tools Development Homepage. http://hpcnc.cpe.ku.ac.th/moin/SCMS_20Cluster_20Management_20Tools

[4] 박은지, 최민, 박동근. 클러스터 환경에서의 부하 분산 시스템 설계 및 구현. Technical Report. Jul, 2003.