

리눅스 기반 시스템 모니터링 프로그램

임동현, 박경민, 최 훈
충남대학교 컴퓨터공학과
e-mail : {dhlm, km-park, hc}@cnu.ac.kr

Linux based System Monitoring Program

Dong-Hyun Lim, Kyung-Min Park, Hoon Choi
Dept. of Computer Engineering, Chung-nam National University

요 약

유비쿼터스 컴퓨팅 환경에서 중요한 요소중의 하나는 안정적인 컴퓨팅 환경을 유지하는 것이다. 그러기 위해서는 사용하고 있는 시스템의 정보를 실시간으로 보여주어 사용자로 하여금 모니터링 할 수 있게 해줄 필요가 있다. 그와 동시에 사용하기 쉬우면서 정확한 기능을 수용하여야 한다. 이런 요구사항을 수렴한 리눅스 기반의 시스템 모니터링 프로그램을 개발하였다 .

1. 서론

컴퓨팅 환경은 소형화와 이동성에 대한 요구를 충족시키는 방향으로 기술의 진보를 이루고 있다. 기술이 날로 발전하고 소형 기기의 점유율 증가와 무선통신 기술이 내장된 단말기의 등장은 언제 어느 곳에서든 인터넷 환경을 손쉽게 접할 수 있도록 해주었다. 이런 기술의 발전은 유비쿼터스 컴퓨팅이라는 목표를 향해 나아가고 있다. 유비쿼터스 환경에서는 안정적인 컴퓨팅 환경을 제공해 주기 위하여 사용하고 있는 기기의 정보, 네트워크 트래픽 정보, 디스크 정보, 메모리 정보 등을 실시간으로 모니터링하고 시기 적절하게 대응 할 수 있는 환경을 제공해 줄 필요가 있다. 그와 동시에 기존의 모니터링 방식에서 오는 거부감을 없앤 사용자에게 친숙한 모니터링 시스템이 필요하다. 이런 요구 사항들을 반영하여 리눅스 기반의 시스템 모니터링 프로그램을 개발하게 되었다.

2. 모니터링 프로그램

2.1. 모니터링 프로그램의 정의

모니터링 프로그램이란 시스템의 자원의 사용 상황을 사용자에게 텍스트나 그래프 등으로 보여주어 시스템의 변화에 적절하게 대처 할 수 있게 만들어주는 프로그램을 의미한다. 모니터링 시스템은 시스템의 현재 상태를 파악하기 위한 목적으로 사용 되고 모니터링을 통해 얻은 정보들은 사용자에게 제공되어 시스템 자원의 활용에 있어서 도움을 준다. 더 진보된 모니터링 시스템은 자동적으로 시스템의 상태에 따라 부족한 자원을 추가 할당하거나 남는 자원을 수집하는 기능을 수행한다.

2.2. 기존의 모니터링 프로그램

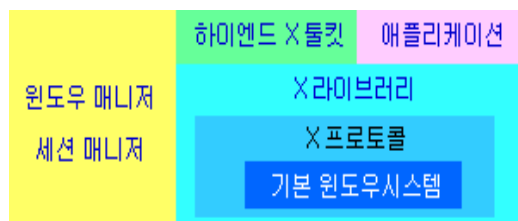
사용자가 가장 쉽게 접할 수 있는 모니터링 프로그램은 윈도우 작업관리자를 들 수 있다. 비스타 이후 버전부터는 가젯 형태의 CPU 측정기가 존재한다. 이것은 항상 시스템의 상태를 확인할 수 있지만 모니터링 프로그램이 제공해야 할 기능들 중에서 많은 부분이 포함되어 있지 않다. 본 논문에서는 보다 사용하기 쉽고 정교한 기능을 제공하는 모니터링 프로그램을 제안한다.

3. UMP

UMP (Ubuntu/UMPC Monitoring Program)는 관리자에게 단순하면서도 꼭 필요한 시스템의 정보를 제공하는 시스템 모니터링 프로그램이다. UMP 는 X-Window 기반에서 QT 를 사용하여 위젯을 통한 GUI 를 구성하였고 libstatgrab 과 libacpi 라이브러리를 사용하여 시스템 모니터링 기능을 구현하였다.

3.1. X-Window

X-Window 는 리눅스에 채용되어 있는 그래픽 환경 기반의 시스템 소프트웨어이다. 리눅스가 각광 받기 시작하면서 보편적인 환경으로 사용되고 있다.



본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다.

[그림 1] X-Window 시스템

X-Window 화면에 나타나는 사용자 인터페이스는 기본 윈도우시스템, X 툴킷, 윈도우 매니저, 세션 매니저 그리고 애플리케이션이 결합된 구조를 가진다. UMP 는 X-Window 프로그램으로 X-Window 의 자원을 이용한다. 자원들은 대표적으로 창(Window), 색상(Color Map), 폰트(font) 등이 있고 이러한 자원들을 X-Window 서버가 전권을 가지고 관리한다.

또한 데스크톱 환경이라 하여 X-Window 를 좀 더 사용자가 편리하게 사용할 수 있도록 윈도우 매니저를 포함하고 여러 종류의 유틸리티(utility)를 package화 하여 사용한다. 제안하는 UMP 는 Gnome Desktop Manage 를 지원한다.

[표 1] X-window 매니저의 종류와 특징

이름	특징
TWM	- 최초의 윈도우 매니저로 다른 윈도우 매니저의 모태
FVWM	- 메모리 사용 최소화 설계, 3D 입체 화면과 가상 데스크탑 지원 - 수정을 통해 새 윈도우 매니저로 기능을 향상시킬 수 있도록 설계
DTWM	- Hewlett-Packard, IBM, Sun Microsystems 가 공동 개발한 유닉스용 상용 GUI 윈도우 매니저 - 유닉스 워크스테이션 표준 운영체제 환경
KDE	- Microsoft 의 윈도우와 구성, 사용법 등이 비슷 - 단순한 윈도우 매니저 역할을 넘어 새로운 데스크탑 환경 구현 목표 - QT 라이브러리를 기반 - kwin 이라는 윈도우 매니저를 가짐
GNOME	- 다른 윈도우 매니저를 띄운 상태에서 실행 가능 - GTK+ 라이브러리를 사용 - metacity 라는 윈도우 매니저를 가짐

3.2. QT 4

QT 는 Trolltech 사에서 개발된 GUI 프로그램 개발을 위한 크로스 플랫폼 위젯 툴킷이다. QT 는 KDE, Qtopia, OPIE 에 이용되고 있으며, C++를 주로 사용하지만 Python, 루비, C, 펄, 파스칼로의 바인딩을 제공하며 그 밖의 많은 플랫폼에서 동작한다. 이는 자유로운 프로그램 환경을 제공한다는 뜻이며 SQL 데이터베이스 접근, XML 처리, 스레드 관리, 단일 크로스 플랫폼 파일 관리 API 등 수많은 함수 및 라이브러리를 제공한다.

QT 는 표준적인 GUI 환경의 지원을 위한 완전한 위젯을 포함하고 있다. 또한 QT 는 QT Designer 라는 유저 인터페이스 디자인 프로그램을 제공하여 UI 를 손쉽게 디자인할 수 있다. QT 의 특징은 다음과 같다.

- C++ 기반
Qt 는 C++을 기반으로 만들어져 있으며 그 구조는 MFC 와 비슷함. 그러므로 기존의 C 개발 방식에 비해서 훨씬 쉽고, 기능 확장이 편리해서 생산성이 높음
- 객체 지향 구조
Qt 라이브러리는 모듈화와, 구성물을 재활용 가능하도록 초점을 맞춰서 설계
- 멀티 플랫폼 지원
MS 윈도우에서 유닉스까지 다양한 플랫폼을 지원하므로, 플랫폼을 바꾸더라도 소스를 수정해야 하는 번거로움이 줄어들음

- Signal & Slot 기반의 동작 방식
Signal & Slot 이라는 QT 만의 이벤트 처리 모델을 사용해서 객체 간의 통신이 가능하고, 복잡한 Callback 함수를 만드는 부담을 줄임

- 국제화
Qt 는 16 비트인 유니코드를 완벽히 지원하며 다양한 ISO 변환 규격과 지역화(Localization)를 지원함. 따라서 인터페이스에 사용되는 모든 문장은 메시지 변환 테이블을 통해 해당 지역 언어로 바꿀 수 있음

- 풍부한 API
Qt 는 250 여개의 C++ 클래스를 지원하며 각 클래스에는 GUI 를 위한 함수와 템플릿 기반의 collection, serialization, 파일, I/O 장치, 디렉터리 관리 및 다양한 종류의 API 를 지원

- 안전성
Qt 는 Trolltech(현 Nokia)에서 모든 것을 관리하므로, 공개된 다른 X 툴킷에 비해 훨씬 안정적이며 Qt 2.0 부터는 CVS 에 등록되어 관리되기 때문에 개발 중인 Qt 소스를 그대로 다운 가능

3.3. libstatgrab 라이브러리

제안 모니터링 프로그램의 개발에는 I-scream 이라는 프로젝트 그룹에서 만든 libstatgrab 이라는 시스템 모니터링 라이브러리를 사용하였다. libstatgrab 은 C 언어로 구성되어 있으며 시스템 자원을 검색하고 통계를 내는 유용한 인터페이스를 제공한다. 특히 CPU 점유율, 메모리 사용률과 상세 정보, 디스크 사용, 프로세스 상태, 네트워크 트래픽 사용, 데이터 입출력 상태 등의 정보를 자세히 알 수 있다.

3.4. libacpi 라이브러리

libstatgrab 에서 제공하지 않는 기능은 H/W 의 전원 사용 정보는 베를린 기술대학교에서 개발된 제작한 libacpi 라는 라이브러리를 이용하였다.

4. UMP 구현

UMP 의 주 개발도구인 QT 는 C++를 기반으로 하는 환경을 제공한다. libstatgrab 과 libacpi 는 C 를 기반으로 만들어져 있지만 QT 는 멀티 플랫폼을 제공하기 위한 도구이기 때문에 C 로 작성된 라이브러리의 사용도 가능하다.

4.1. 패키지 구성

UMP 패키지 내부의 파일들은 <표 3>과 같다. UMP 클래스를 이루는 헤더파일과 소스파일, 각 모니터링 요소들의 임계값을 기록한 파일 등으로 구성되어 있다.

[표 2] UMP 패키지 구성

파일 명	설명
Makefile	UMP 를 컴파일 하기 위한 Makefile
ump.cpp	QT 로 구현된 프로그램 본체 구현부
ump.h	QT 와 관련된 함수 선언
moc_u-ump.cpp	QT 의 Meta Object Compiler 구현부
HELP.html	라이센스 정보가 기술된 웹문서
SETxxx	xxx 는 각 기능의 Threshold(임계점)을 저장하는 파일

4.2. Class 분석

ump.cpp 는 X-Window 에서 사용자에게 보여지는 GUI 의 본체를 구현하고 있다. 구현된 class 들은 UMP 를 실행하였을 때 X-Window 에 뿌려지는 위젯의 본체를 구성하였다. 시스템의 기본 정보부터 각 시스템 구성의 자원에 대한 상황을 실시간으로 보여주며 그 그래프와 상태바(progressbar), 텍스트 등을 통해서 시스템 관리자에게 필요한 자원 정보의 변화를 보여준다. 그래프는 실시간으로 그려지며 20 초마다 클리어 된다. 특히 네트워크 트래픽 모니터링은 각 네트워크 인터페이스 별로 선택적으로 상태정보 변화를 모니터링 할 수도 있다. 각 모니터링 대상 별로 사용자가 지정함 임계값을 넘어갈 경우 GUI 화면의 해당 자원에 경고등이 들어와서 사용자가 쉽게 인지할 수 있도록 구현하였다.

[표 3] UMP 클래스의 함수 목록

함수명	설명
UMP()	UMP 객체의 생성자.
void show_help()	help 버튼 클릭시 UMP_HELP 객체를 생성하여 보여줌.
void show_setting()	setting 버튼 클릭시 UMP_SETTING 객체를 생성하여 출력함
void get_systeminfo()	시스템의 정보를 가져옴.

void main_info_frame()	hostname uptime 에 대한 label 표시
void main_info_value()	hostname uptime 의 값을 표시
void cpu_info_frame()	cpu 사용 정보 에 대한 label 표시
void cpu_info_value()	cpu 사용 정보의 값을 표시
void process_info_frame()	process 정보에 대한 label 표시
void process_info_value()	process 정보에 대한 값을 표시
void mem_info_frame()	Memory 정보에 대한 label 표시
void mem_info_value()	Memory 정보에 대한 값을 표시
void disk_info_frame()	Filesystem 정보에 대한 label 표시
void disk_info_value()	Filesystem 정보에 대한 값을 표시
void network_info_frame()	Network 정보에 대한 label 표시
void network_info_value()	Network 정보에 대한 값을 표시
void battery_info_value()	Battery 정보에 대한 값을 표시
int get_stats()	시스템의 정보를 선언한 구조체에 저장함
void load_setting_file();	SETTING 파일을 불러옴.
void write_setting_file();	SETTING 파일을 저장함..
char size_conv()	각 정보단위를 Mb,byte 단위로 표기하기 위한 함수

4.3. Signal & Slot 이벤트 처리

UMP 의 화면에 표시되는 관리대상의 상태정보는 매 1sec 마다 갱신된다. 이것은 QT 에서 제공하는 Signal & Slot 이벤트 처리 방식을 이용하여 구현하였다. timer 가 매 1sec 마다 connect 함수의 이벤트를 호출하게 되면 객체 내부의 slotTimer()함수가 호출된다. slotTimer()함수는 UMP 의 시스템 정보 값을 갱신하는 역할을 담당하도록 되어있다.

[표 4] Signal & Slot 구조

```
timer = new QTimer(this);
connect(timer,SIGNAL(timeout()),
        this,SLOT(get_systeminfo()));
connect(timer,SIGNAL(timeout()),
        this,SLOT(slotTimer()));
timer->start(1000);
```

connect 함수는 timer 가 시간이 지나면 get_systeminfo()와 slotTimer()를 수행하게 하는 대표적인 함수이다.

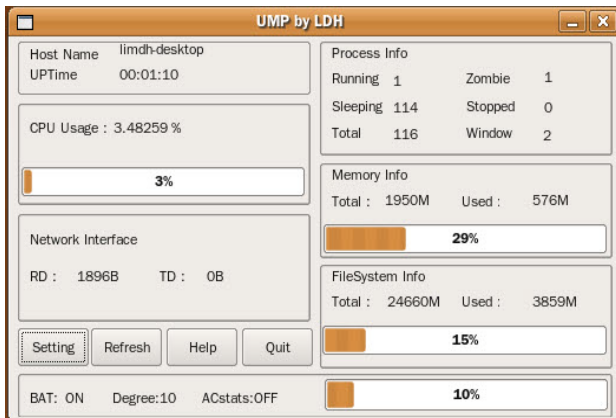
5. 설치 및 실행

UMP 는 QT 라이브러리와 libstatgrab 그리고 libacpi 구성요소의 설치를 필요로 한다. 터미널 모드에서 apt-get 을 통한 수동 설치는 의존성 파일들이나 기타 호환상의 문제가 발생할 수 있기 때문에 우분투에서 제공하는 시냅틱 패키지 관리자를 통한 설치를 권장한다. 이 패키지 관리자는 설치하고자 하는 패키지의 의존성 파일들을 자동으로 검색하여 설치해주고 시스템에 맞는 최적화된 옵션으로 설정을 해준다.

필수 라이브러리의 설치 후에 UMP 패키지를 설치하면 UMP 를 사용할 수 있다.

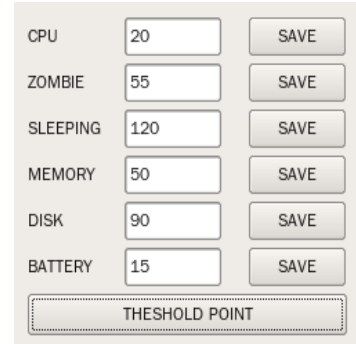
6. 동작 시나리오

UMP 를 실행하면 [그림 2]와 같은 모니터링 화면이 출력된다.



[그림 2] UMP 실행 화면

- main info frame
시스템의 hostname 과 동작시간을 보여준다.
- cpu info frame
cpu 사용률을 보여준다. 다중 코어의 경우에는 각각의 사용률이 아닌 전체 사용률을 보여준다.
- network info frame
모니터링 되고 있는 네트워크 인터페이스의 이름과 RX, TX 의 크기를 보여준다.
- process info frame
시스템 상에서 동작중인 모든 프로세스의 개수가 나오고 상태 별로 Running, Sleeping, Stopped, Zombie, window, Total 로 나누어 분류하여 보여준다.
- memory info frame
메모리의 전체 크기와 사용 중인 크기와 사용 중인 크기가 텍스트와 progress bar 로 보여준다.
- disk info frame
하드 디스크의 전체 크기와 사용량을 텍스트와 progress bar 로 보여준다.



[그림 3] 임계값 설정

[그림 3]은 관리 대상의 임계값을 설정하는 화면이다. 설정창에서 설정된 임계값을 기준으로 UMP 는 각 관리대상의 상태정보를 수집하고 비교하여 임계값을 넘는 관리대상은 UMP 실행화면에 붉은 색으로 처리하여 사용자에게 통보한다.

7. 결론

유비쿼터스 컴퓨팅 환경이 도래하면서 다양한 모바일 디바이스에서 리눅스가 사용되고 있다. 모바일 디바이스는 제한적인 컴퓨팅 파워와 자원을 가지기 때문에 시스템의 정보를 확인하고 적절한 조치를 취해야하지만 사용자에게 친숙한 시스템 모니터링 프로그램이 부족한 상황이다.

본 논문에서 제안하는 UMP 프로그램은 QT 기반으로 시스템 상태정보를 제공하는 libstatgrab 라이브러리와 디바이스의 전원 상태정보를 제공하는 libacpi 라이브러리를 이용하여 개발되었으며, 사용자에게 친숙한 GUI 를 통해서 각 시스템의 관리대상의 사용 정보를 제공하고 적절한 임계점을 설정하여 임계값이상의 사용률을 보이는 관리대상의 위험성을 사용자에게 통보할 수 있는 기능을 가진다.

참고문헌

- [1] 박충범, 유용덕, 최훈, "웨어러블 컴퓨터에서 Self-Configuration 을 위한 Monitoring 기법", 한국정보처리학회 제 26 회 추계학술발표회 논문집(하), pp.757-760, 2006.11.
- [2] Choong-Bum Park, Ki-Jeong Kwon, Yong-Duck You, Hoon Choi and Yong-Hwan Jung, "The Service Management System for Ubiquitous Environment," International Conference On Next-Generation Computing, Nov. 2007.
- [3] Nokia, "Product-A cross-platform application and UI framework," <http://qt.nokia.com/products>.
- [4] 황선엽, 원영암, 유용덕, 최훈, "웨어러블 컴퓨터를 위한 전력관리 미들웨어 성능 측정", 2007 한국컴퓨터종합학술대회 논문집, Vol.34, No.1(A), pp.254-255, 2007.06. [정보통신부 선도기반 기술사업]
- [5] Max Berger, "Introduction to the ACPI specification and its implementation in the Linux Operation System," <http://max.berger.name/als/als.pdf>