

# 무선 기반의 실시간 부하 모니터링 시스템

## Real-time load monitoring system based mobile

박홍진(Park Hong Jin)<sup>1)</sup>

### 요 약

컴퓨터 시스템과 IT 기술의 폭발적인 발전으로 인해 컴퓨터 시스템은 점점 복잡해지고 있다. 복잡한 시스템을 보다 효율적으로 관리하기 위해서는 시스템 부하 모니터링 기술이 필요하다. 본 논문은 기존의 관리자가 부하 정보 관련 명령어를 이용하거나 유선기반의 특정 모니터링 프로그램을 수행하는 방식에서 무선 기반의 실시간 부하 모니터링 시스템을 구현한다. 본 논문에서 구현된 부하 모니터링 시스템 기존 방식보다 실시간적이면서 시공간에 자유롭게 부하 정보를 확인 할 수 있는 장점이 있다.

### Abstract

The computer system is gradually complicated because of explosive growth of computer system and IT technology. For efficient management of complicated system, system load monitoring technology is essential. In this paper, we is implemented the real-time load monitoring system based mobile than the traditional methods that are used load-related commands in system manager or specific load application programs based connected wire. Advantage of implemented load monitoring system in this paper is freely saw load information in time and space though mobile method like as PDA, cellular phone, more than traditional methods in time and space.

논문접수 : 2004. 11. 29.

심사완료 : 2004. 12. 15.

1) 정회원 : 상지대학교 컴퓨터정보공학부

“이 논문은 2003년도 상지대학교 교내 연구비 지원에 의한 것임“

## 1. 서론

컴퓨터 시스템 환경이 복잡해지면 복잡해질수록 시스템에서 발생하는 문제는 다양하게 발생할 수 있다. 복잡한 시스템을 효율적으로 관리하기 위해서는 전반적인 시스템의 성능을 관리해 주는 시스템 성능 관리가 무엇보다 중요하다. 시스템의 성능 관리는 먼저, 현재 시스템의 가용성과 상황을 명확하게 파악함으로써 향후 시스템의 부하(load)가 가중될 때 시스템의 용량 계획에 기준으로 사용함으로써 불필요한 투자를 감소시킬 수 있는 장점을 제공한다. 또한, 시스템의 성능 관리에서 시스템이 지속적으로 안정적인 상태로 운영되기 위해서는 시스템의 자원이 이상이 없는지를 실시간적으로 감시하는 모니터링 기술이 필요하다. 예를들어, 시스템에 어떠한 장애가 발생하려는 징후가 보이거나 이미 장애가 발생했다면 신속한 조치를 취하는 것이 무엇보다도 급한 일이며 이러한 장애 발견과 신속한 조치를 취하기 위해서는 무엇보다도 장애 현황을 즉각적이고, 실시간으로 부하 정보를 파악하는 기술이 선행되어야 한다[1].

시스템 부하 모니터링 정보는 현재 시스템에서 실행되고 있는 각종 프로세스들과 최근 CPU 평균 부하율, 현재 사용한 메모리 관련 정보, 스왑 메모리 정보, 현재 로그인한 사용자에 대한 정보, 파일 시스템에 대한 정보, OS 정보등 다양한 정보를 제공해야한다[2][3][4].

기존의 시스템 성능 모니터링은 시스템 관리자가 성능 관련 명령어를 이용하여 텍스트 기반의 모니터링 하는 방법이나 성능 모니터링이 가능한 어플리케이션 이용하여 모니터링 하는 방법, 웹 기반에서 모니터링 하는 방법등 주로 유선 인터넷 중심의 모니터링 방식으로 처리된다. 이는 언제 어디서나 필요로 하는 정보를 가져오는 인터넷 서비스를 수행하는 데 한계를 지니고 있다. 더욱이 장애는 언제 어디서 발생할지 알 수 없기 때문에 전체 시스템 관리 대상에서 발생하는 장애를 시간과 장소에 무관하게 관리자가 파악할수 있는 무선 기반의

모니터링 기술이 필요하다.

무선 인터넷이란 사용자가 무선단말기로 이동 중 무선망을 통하여 인터넷 서비스에 접근하고 정보를 제공받을 수 있도록 하는 환경과 기술이다. 본 논문은 무선 인터넷을 통해 실시간적으로 시스템 성능을 모니터링할 수 있는 시스템 구현을 목적으로 한다. 구현된 시스템은 무선 인터넷으로 정보를 실시간적으로 얻기 때문에 무엇보다도 시공간적으로 유선 인터넷 보다 자유롭게 성능 정보를 모니터링 할 수 있는 장점이 있다.

본 논문의 구성은 다음과 같다. 2장에서는 성능관리의 중요성과 기존 부하 모니터링 방법을 설명한 한 후 3장과 4장에서 무선 기반 실시간 부하 모니터링 시스템의 설계 및 구현에 대해서 설명하고 끝으로 5장에서는 결론을 맺는다.

## 2. 성능 관리의 중요성과 기존 성능 모니터링 방법

시스템의 효율적인 성능 관리를 통해 시스템의 성능 개선과 자원 최적화, 향후 요구에 대한 적절한 계획을 제공한다는 점에서 ROI(Return of Investment) 기대 효과 높일 수 있는 기술이다. 효율적인 시스템의 성능 관리를 위해서는 시스템의 관리 대상을 확정하고 이에 따라 실시간적인 성능 모니터링 기술이 필수적이다. 성능 모니터링 기술은 다음과 같은 장점을 제공한다. 첫째, 시스템에서 사용하지 않는 불필요한 작업에 대한 정보를 제공할 수 있다. 이는 불필요한 데몬 프로세스 사용에 대한 정보를 검사할 수 있고, 이 작업은 보안 문제에도 연관될 수 있다. 둘째로 시스템 자원에 대한 효율적인 관리를 제공한다. 전체 시스템의 성능을 저하시키지 않도록 특정 자원의 병목현상에 대한 관리를 할 수 있으며 시스템의 작업 부하를 관리할 수 있는 자료를 제공한다. 셋째로 향후 시스템 확장시 추가적인 비용을 최소화 시킬 수 있다.

기존의 시스템 성능 모니터링 기법은 첫째,

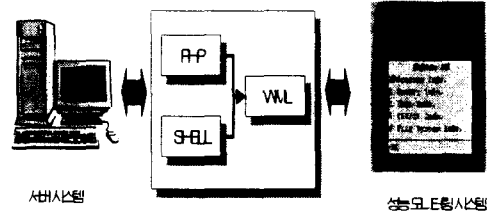
시스템 관리자가 성능 관련 명령어를 이용하여 텍스트 기반의 모니터링 하는 방법이다. 시스템 관리자가 ps나 uptime, prstate등의 명령어를 사용하여 프로세서의 부하 정보를 얻거나, sar나, wmsat등의 명령어를 사용하여 메모리에 관한 정보를 얻는 방법이다. 둘째로, 특정 어플리케이션을 사용하거나, 웹 기반의 모니터링 하는 방법이다. Webalizer[5]는 리눅스에서 가장 많이 사용되는 웹 서버 로그 분석 프로그램이며, Wisensniffer[6]는 네트워크 패킷을 가로채 로그 분석하며, MRTG(Multi-router Traffic Grapher)[7]는 네트워크 링크간 트래픽 부하량을 측정하는 도구로 이용하고 있다. 이러한 기존 성능 모니터링 기법은 시스템 관리자가 성능 관리 명령어를 잘 인식하고 있어야 하며, 유선으로 연결된 컴퓨터 상에서 성능 관리가 수행됨으로 무선 기반의 성능 모니터링 기법에 비해 시간과 장소에 구애를 받는다.

### 3. 무선 기반 부하 모니터링 시스템 구성

무선 인터넷은 무선 통신을 이용한 인터넷 접속. 이동 전화, 개인 휴대 정보 단말기(PDA) 등의 무선 장비나 무선 LAN, 블루투스 같은 무선 시스템을 통해 인터넷 서비스를 제공하는 것으로, 시간과 장소에 구애받지 않고 언제 어디서나 인터넷 서비스를 사용할 수 있는 장점이 있다. 인터넷에 이동통신의 이동성(mobility)을 접목하기 위해서 무선망과의 연동이 필요하며 WAP(Wireless Application Protocol) 방식과 ME(Mobile Explorer) 방식의 표준이 제시되고 있다.

본 논문에서는 MS사에 제안한 ME 방식보다 WAP 방식을 이용하고 있는데 이는 국제 무선 인터넷 표준 프로토콜인 WAP(Wireless Application Protocol)은 WAP포럼에서 정의하고 규정한다.. WAP 방식은 세계적으로 사용자 면에서 가장 많은 수를 차지하고는 있지만 HTTP를 지원하지 않는 기존 유선인터넷 프로토콜과의 비호환성문제로 WAP 게이트웨이(Gateway)가 필요하며, 그로 인한 비용의 추가

등의 문제점이 있다. ME 방식은 브라우저의 오버헤드가 크고, 소스를 공개하지 않기 때문에 브라우저에서 지원하지 않는 파일을 이용한 서비스가 불가능한 단점이 있다[8][9][10].



[그림 1] 무선 기반 성능모니터링 시스템의 구성

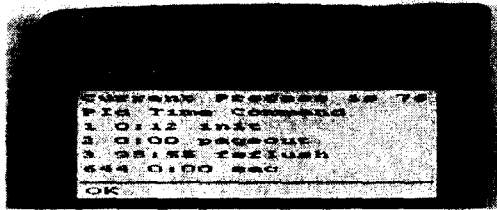
[그림 1]는 본 논문에서 구현하고자 하는 무선 기반 성능모니터링 시스템의 전반적인 구성을 설명하고 있다. 서버 시스템은 인터넷을 통해 일반 사용자 제공하는 시스템을 의미한다. 서버가 올바른 작동하는지 감사하기 위해서는 현재 사용중인 프로세스 정보, 현재 메모리, 현재 사용자 정보등 다양한 정보가 모니터링되어야 한다. 본 논문에서는 서버에서 성능 정보를 제공하기 위해 PHP와 SHELL, WML을 연동하여 설계하였다. 서버 관리자는 무선 단말기를 통해 시공간에 관계없이 서버 시스템에 연결후에 서버 부하 정보를 모니터 할 수 있다.

## 4. 무선기반 부하 모니터링 시스템의 구현

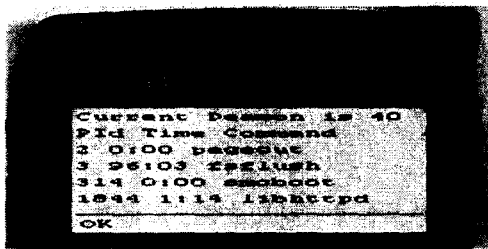
### 4.1 프로세스 정보

프로세스 정보는 크게 현재 수행 중인 프로세스 정보와 현재 수행 중인 데몬 프로세스 정보 정보가 포함되어 있다. 현재 수행 중인 프로세스 정보는 [그림 2]와 같다. 현재 시스템에 수행중인 프로세스 개수의 합이 76개임을 나타내고 있다. PId은 프로세스 번호이다. 프로세스를 강제로 Time은 프로세스 명령을 실행하는

데 현재까지 사용된 CPU 누적 시간이다. Command는 프로세스 이름이다. [그림 3]는 현재 수행 중인 데몬 프로세스를 정보를 나타내고 있다. 현재 데몬 프로세스가 40개임을 나타내고 있으며 나머지 정보는 [그림 2]와 같다.



[그림 2] 현재 수행 중인 프로세스 정보

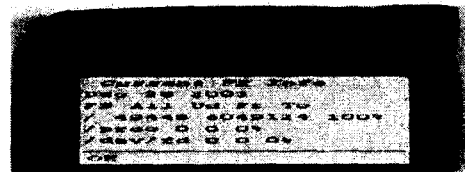


[그림 295] 현재 수행 중인 데몬 프로세스 정보

#### 4.2 메모리 정보

메모리 정보는 크게 현재 사용한 메모리 정보와 스왑 메모리 정보가 포함되어 있다. 메모리 정보는 시스템 내부의 디스크를 사용한 량과 사용가능한 디스크량을 실시간적으로 측정함으로써 디스크 풀 현상을 미리 예방할 수 있다. 디스크 풀 현상이 발생하게 되면 시스템 운영에 치명적인 오류가 발생하게 되며 웹 서비스를 제공할 수 없다. 또한, 현재의 스왑 메모리(swap memory)량은 시스템의 작업과 프로세스가 증가할수록 현재 사용가능한 스왑 메모리의 양이 줄어들게 된다. 스왑 메모리 양이 적을 경우 작업 처리 속도가 느려져 웹서비스 속도도 저하된다. 현재의 프리 메모리(free memory)는 시스템의 작업과 프로세스가 증가

할수록 현재 사용가능한 메모리의 양이 줄어들게 된다. 메모리 양이 적을 경우 작업 처리 속도가 느려져 서버 속도도 저하된다. [그림 4]은 현재 사용한 메모리 정보를 나타내고 있다. FS는 현재 파일 시스템의 설치 지점을 나타낸다. 또한 Ail은 현재 파일 시스템의 전체 디스크 용량과 Ud는 사용한 용량을 K로 나타낸 것이며 Fr은 사용 가능한 용량을 K로 나타낸 것이고, 마지막으로 To는 사용 가능한 용량을 백분율로 나타난 것이다. [그림 5]은 현재 시스템의 스왑 메모리 정보를 나타내고 있으며, Path은 파일 시스템의 장치를 나타내며 나머지는 [그림 4]와 같다.



[그림 296] 현재 사용한 메모리 정보

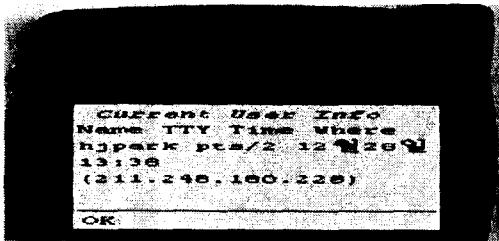


[그림 297] 현재 스왑 메모리 정보

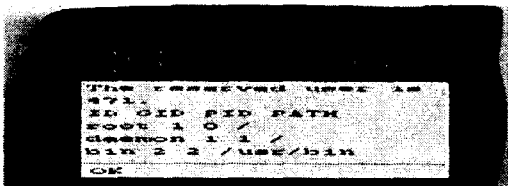
#### 4.3 사용자 정보

사용자 정보에서는 현재 로그인 한 사용자에 대한 정보, 현재 서버에 등록된 사용자 정보와 현재 서버에 등록된 그룹 정보가 있다. [그림 6]은 현재 로그인한 사용자 정보를 나타내고 있다. Name은 로그인한 사용자 이름이며, TTY는 사용자가 접속한 단말기 이름이고, Time은 사용자가 로그인 한 시간, Where는 사용자가 로그인 한 장소(site)을 의미한다. [그림 7]은 현재 서버에 등록된 사용자 정보를 의미한다. [그림 7]에서 현재 등록된 사용자는 총 471명이고, ID, GID, PATH는 각각 사용자 이름과 그룹 번호, 프로세스 번호, 경로를 나타낸

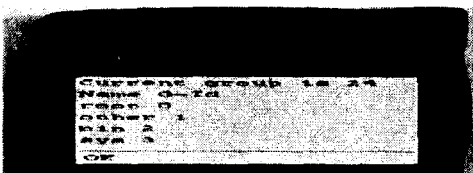
다. [그림 8]은 현재 서버의 등록된 그룹 정보를 나타낸다. [그림 8]에 나타나 있듯이 현재 웹 서버에서는 총 24개의 그룹이 있고, 세부적인 목록은 [그림 8]에 나타나 있다.



[그림 298] 현재 로그인한 사용자에 대한 정보



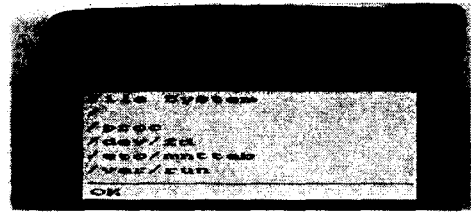
[그림 299] 현재 서버에 등록된 사용자 정보



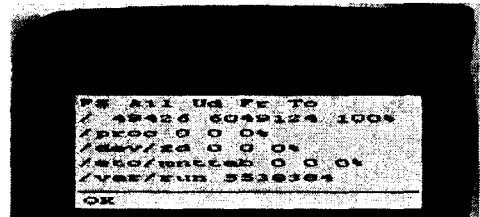
[그림 300] 현재 서버에 등록된 그룹 정보

#### 4.4 파일 시스템 정보

파일 시스템 정보는 파일 시스템의 구조에 대한 정보와 파일 시스템의 정보를 포함되어 있다. [그림 9]은 파일 시스템 구조의 대한 정보를 나타내고 있다. [그림 10]는 파일 시스템의 정보를 나타낸다. FS는 파일 시스템 설치 지점을 나타내며 나머지 정보는 [그림 4]과 같다.



[그림 301] 파일 시스템 구조의 대한 정보

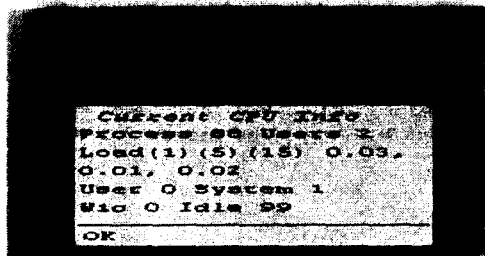


[그림 302] 파일 시스템의 정보

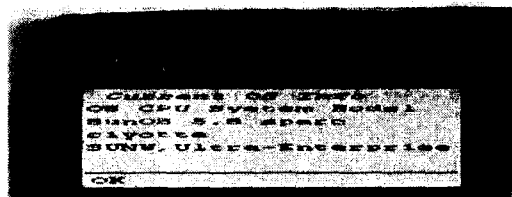
#### 4.5 CPU와 OS 정보

CPU정보는 현재 실행중인 프로세스들과 사용자 수, CPU 사용률, CPU의 실행 큐의 크기를 통해 CPU에 부하 정도를 판단할 수 있다. 실행 큐의 평균 작업수는 시스템의 전체적인 부하정도를 나타낸다. 평균 작업수가 높을 수록 최근의 시스템이 실행한 프로세서의 수가 많았다는 것을 의미하면 메모리와 CPU 사용률이 증가하게 된다. 평균 작업수가 많을 수록 시스템의 부하가 증가하게 되며 사용자에게 제공하는 서비스의 속도가 줄어들게 된다. 또한, 큐에 적재되는 작업 개수가 지속적으로 증가한다면 단위 시간에 처리해야할 프로세스가 CPU 파워 부족으로 인해 처리하지 못하고 큐에 지속적으로 쌓이고 있음을 의미한다. [그림 11]은 부하 정보를 나타낸다. 현재 시스템의 수행중인 프로세스 개수는 88이고 2명이 시스템에 접속하여 사용중에 있으며, 최근 1분, 5분 15분간 수행 한 작업 큐의 평균 작업수는 Load(1)(5)(15)의 0.03, 0.01, 0.02개 이다. [그림 12]은 OS에 대한 정보를 나타낸다. 현재 시스템은 SUNOS 5.8, Sparc 프로세서를 사용하고 있으며, 모델명은 Ultra-Enterprise임을 알 수

있다.



[그림 303] CPU 정보



[그림 304] OS 정보

### 5. 결론

인터넷 사용자의 폭발적인 증가는 서버 시스템의 중요성을 부가 시키며, 정보 통신 기술의 발전은 무선 인터넷의 중요성을 점점 부가시키고 있다. 본 논문은 효율적인 성능관리를 위해 서버의 부하 정보를 실시간 적이면서 즉각적인 정보를 얻기 위해 무선 기반 서버 부하 모니터링 시스템을 제안하였다. 본 논문에서 제안한 모니터링 시스템은 기축 유선에서 제공하는 서비스보다 시간과 장소에 구애받지 않고 언제 어디서나 인터넷 서비스를 사용할 수 있는 장점이 있으며, PDA등을 통해 WAP 프로토콜을 이용하여 서비스 제공 받는다. 본 논문에서 개발된 시스템의 정보는 프로세스, 메모리, CPU 부하 정보등이 제공되며 무엇보다도 무선 인터넷을 이용한 부하 정보를 제공하는 잇점이 있다.

### 참고 문헌

- [1] M. Lindermeier, "Load management for distributed object-oriented environments", International Symposium on Distributed Objects and Applications 2000.
- [2] SAS Application Performance Monitoring for UNIX,

[www.sas.com/partners/directory/hp/sasapp.pdf](http://www.sas.com/partners/directory/hp/sasapp.pdf)

- [3] System Performance Monitoring, [www.sun.com/columns/adrian/column1.html](http://www.sun.com/columns/adrian/column1.html).

- [4] Specifying Load Monitoring Options,

[www.csse.monash.edu.au/cluster/enfman80/x3309.htm](http://www.csse.monash.edu.au/cluster/enfman80/x3309.htm).

- [5] webalizer, [www.webalizer.com](http://www.webalizer.com).

- [6] wisensniffer, <http://www.wiselog.com/clm0s1.php>.

- [7] Tonia Oetiker and Dave Rand, "MRTG: Multi Router Traffic Grapher", <http://www.mrtg.org>.

- [8] <http://www.wapforum.com/>.

- [9] Huw Evans, "Getting Started with WAP and WML", SYBEX, 2001.

- [10] 김성환, 양석호저, "모바일 자바 프로그래밍", Prentice Hall, 2002.