韓國 컴퓨터情報學會 論文誌
第10卷 第4號, 2005. 9.

2005-10-4-1-16

웹기반 서비스 감시 시스템의 구현

조 승 한*

Implementation of Web-based Service Observation System(SOS)

Seung-Han Cho *

요 약

웹 서비스를 포함한 다양한 서비스를 제공하는 대학, 회사의 전산소는 비용 관계상 결함포용(Fault-Tolerant) 시스템이 아닌 단일 서버, 단일 네트워크 장비로 구성되어 있다. 이렇듯 이중화로 구성되어 있지 않은 시스템의 경우 웜 바이러스 등과 같은 다양한 이유로 장애가 발생할 수 있으므로 이러한 장애를 능동적으로 감지할 수 있는 기술이 필요하다. 본 논문은 다양한 서비스에 대해 장애가 발생한 시스템을 감시하는 시스템을 설계하고 구현하며 관리자의 편의성을 위해 웹 기반 시스템으로 구현하는 기술을 소개한다. 이러한 시스템을 사용하여 시스템 관리자는 사용자의 장애 신고에 의지하지 않고 장애가 발생한 서비스를 이메일, SMS 등을 통해 보고받아 즉각적인 장애 조치를 취할 수 있다.

Abstract

Computer center of university or company manages many non fault-tolerant servers and network devices to spare expenses. Because a service fault occurs sometimes by worm virus, system bug etc, we need a technique to detect it for continuing service. This paper introduces design and implementation of the system to observe many heterogeneous services, and web-based interface improving convenience of system manager. A system fault is reported to system managers via email or SMS by introduced service observation system, not service user. Then system managers can recover the system fault by this notification and minimize a fault period.

▸ Keyword : 서비스 감시(Service Observation), 네트워크 감시(Network Observation)

• 제1저자 : 조승한
• 접수일 : 2005.07.15, 심사완료일 : 2005.09.05
* 용인송담대학 컴퓨터게임정보과 조교수

4. 자원 부족에 기인한 장애를 위해 시스템에 대한 성능 감시 및 보고를 수행해야 한다.

Ⅰ. 서 론

오늘날 대학, 기업에서 운영하고 있는 서버는 HTTP 서버를 비롯하여 DNS 서버, 메일 서버, FTP 서버, 데이터베이스 서버 등 상이한 기능을 갖는 다양한 서버들로 구성되어 있으며 외부와 네트워크에 연결되어 자신에게 부여된 서비스를 제공하고 있다. 이러한 전산 장비는 하드웨어 결함, 시스템 자원의 부족, 하드웨어/소프트웨어의 버그, 웜 바이러스 등의 다양한 이유로 장애가 발생할 수 있는데 이러한 장애에 대해 관리자는 수시로 점검하여 장애가 발생하지 않는 환경을 조성하고 있다. 하지만 이러한 노력에도 불구하고 다양한 원인에 의해 장애가 간헐적으로 발생할 수 있는데 장애가 발생한 경우 관리자는 이 사실을 사용자의 전화 통지와 같은 적극적인 사용자 보고와 우연에 기인한 관리자의 서비스 실패 경험에 의해 인지하여 복구 조치를 취하고 있다. 서버가 지속적인 서비스를 제공하기 위해서는 장애가 발생한 시점에서부터 복구가 완료됐을 때까지의 시간을 최소화시키는 것이 필요한데 이것을 결정짓는 주요 요인은 바로 장애 사실을 관리자가 인지하게 된 시점인 것이다.

시스템 모니터링 시스템이란 서버, 네트워크 장비 등 다양한 전산 장비의 서비스 상태를 주기적으로 검사하여 서비스의 장애 발생시 장애 내용을 즉각적으로 관리자에게 알려주는 시스템을 의미한다. 즉, DNS 서버, HTTP 서버, 데이터베이스 서버 등 다양한 서버들과 스위치, 라우터 등의 네트워크 장비로 구성되어 있는 기관의 전산 장비에 대해 장애가 발생했을 경우 이를 관리자에게 즉각적으로 보고하여 복구 조치가 이루어질 수 있도록 하기 위한 시스템인 것이다. 이 시스템이 갖춰야한 주요 구성 기능은 다음과 같다.

1. 주기적 검사 또는 시스템의 서비스 상태 보고를 통해 현재의 장애 여부를 실시간으로 감지하여야 한다.
2. 검사 대상 시스템은 ping 에 의한 네트워크 통신 유무에 의한 검사뿐 아니라 데이터베이스 서비스를 포함한 다양한 서비스가 검사되어야 한다.
3. 장애가 감지되면 이메일, 핸드폰 단문 메시지(SMS), 백그라운드 프로그램의 활성화 등 다양한 방법을 통해 관리자에게 보고 되어야 한다.

Ⅱ. 기존 연구

2.1. 독립 어플리케이션 방식

상용 프로그램으로 국내 탭스랩에서 제작한 Server Observer[1] 프로그램이 있으며, 외국산으로는 WoodStone의 Server Alive[2], ManageEngine 의 OpManager[3] 프로그램 등이 있다. 다음 (그림 1)은 국내에서 가장 범용성을 갖고 사용되고 있는 Server Observer 프로그램 화면이다.

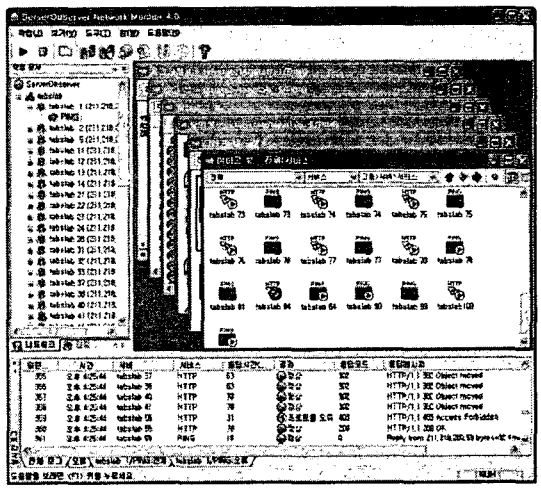

그림 1. Server Observer 프로그램 실행 화면
FIG 1. Server Observer Program Window

이 방식들은 위에서 제시한 시스템 모니터링 시스템 기능 중 4번째 항목인 성능 감시 및 보고 기능을 제외한 다른 모든 기능들을 충족하고 있다. 하지만 웹 기반이 아닌 독립 어플리케이션 방식으로 이 프로그램이 실행중인 시스템외의 다른 컴퓨터에서는 현재의 시스템 감시 결과를 조회할 수 없는 한계를 갖고 있다. 따라서 이러한 독립 어플리케이션 방식은 시스템별로 관리자가 배정되어 있거나 여러 관리자가 동시에 감시하는 현재의 시스템 관리 상황에서 장애 발

생에 따른 장애 내역 조회나 통계 정보 추출에 많은 불편함을 갖는다. 또한 Server Observer 제품만이 성능 정보를 수집 관리하고 있는데 이러한 성능 정보를 이용하여 일정 수준 이상의 성능 결함을 감시하고 보고하는 체제는 현재 구현되고 있지 않은 상태이다.

2.2 웹기반 방식

웹 기반 방식으로는 MRTG(Multi Router Traffic Grapher)[4]가 현재 사용할 수 있으며 무료로 설치할 수 있다. 다음은 네트워크 전용 회선에 대한 PING 응답 시간과 라우터의 성능을 (그림 2)에서 보여주고 있다.

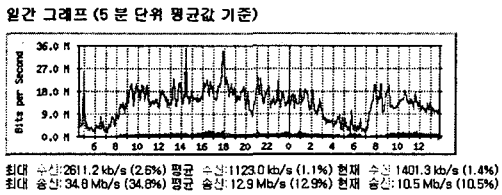

그림 2. MRTG 그래프
Fig 2. MRTG graph

그러나 이 방식은 HTTP, DNS 등의 서비스에 대한 감시가 아니라 네트워크의 연결성에 중심을 두어 PING 응답 시간 및 SNMP 를 이용한 시스템의 성능을 모니터링[5]하고 있는 시스템이다. 따라서 위에서 소개한 바와 같이 엄격한 의미에서의 시스템 모니터링 시스템이라 불릴 수 없는 방식이다. 하지만 이 방식은 웹을 이용해 차트로 표현한 방식으로 사용자로 하여금 시스템의 상태를 좀 더 보기 편하게 구성되어 있다.

따라서 이 논문에서는 이러한 기존의 방식들의 장점을 결합하여 다양한 프로토콜에 의한 서비스 검사를 수행하며 웹을 사용해 관리하여 관리자에게 많은 편리성을 제공하고 성능 검사를 수행해 성능 장애에 대해 보고하는 시스템을 설계 구현한다.

III. SOS 설계 및 구현

3.1 설계

이 시스템은 (그림 3)과 같이 7 개의 구성 요소를 갖는다. 먼저 서비스 감시 모듈은 DNS, HTTP, FTP, SMTP, POP3 등 다양한 네트워크 서비스와 데이터베이스 서버가 정상적으로 작동하고 있는지를 검사하기 위한 모듈이다. 이것은 등록된 시스템에 대해 하나의 멀티스레드로 서비스를 감시하도록 구성되어 있어 다수의 시스템에 대해서도 병행적으로 실시간 처리가 가능하다. 두 번째 PING 감시 모듈은 ICMP 를 사용하는 PING 프로그램과 같은 방식에 따라 시스템이 네트워크에 연결되어 있고 프로토콜 스택이 동작중인지를 감시하기 위한 모듈이다. 이것은 단순히 허브, 스위치, 백본, 라우터와 같은 네트워크 노드의 동작 여부를 확인하기위한 용도 뿐 아니라 응용 프로토콜에 대한 서비스 장애 시 시스템이 네트워크에 연결되어 동작중인지 여부를 감시하기 위한 모듈이다.

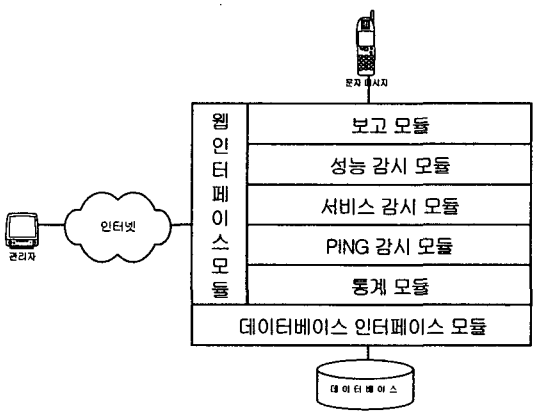

그림 3. 시스템 구성도
Fig 3. System Structure Diagram

세 번째로 성능 감시 모듈은 시스템의 성능을 감시 보고하기 위한 모듈이다. 이 모듈의 특징은 관리자가 지정된 수준의 성능 임계치에 시스템이 도달한 경우 성능 장애로 관

리자에게 보고하도록 구성되어 있다. 네 번째로 보고 모듈은 서비스의 장애 및 성능 장애 현상을 이메일 또는 핸드폰 문자 메시지(SMS)를 이용하여 관리자에게 보고하는 모듈이다. 다섯 번째 통계 모듈은 데이터베이스에 수집되어 있는 상태 정보를 하루 단위의 통계 정보로 변환 저장하는 모듈이다. 이것은 상태 정보가 일주일 분이 저장되는데 상태 정보의 끝없는 증가를 막고 하루 단위의 요약 정보를 저장하기 위함이다. 여섯 번째 데이터베이스 인터페이스 모듈은 수집된 서비스 감시 데이터 및 성능 데이터를 저장 관리하기 위한 모듈이다. 마지막으로 웹 인터페이스 모듈은 웹을 이용한 환경 설정 및 현재의 감시 결과를 조회하기 위한 모듈이다. 여기에는 마지막으로 수행된 감시 결과의 조회 뿐 아니라 현재의 검사 상황을 즉각적으로 보여주기 위한 자바와의 통신 모듈이 포함되어 있다.

서비스 감시 결과에 따라 다음과 같은 상태 정보를 관리 운용 보고한다.

표 1. 장애 상태 설명
Table. 1 Fault State Description

장애상태	설명	비고
1급장애	통신 프로토콜 오류, 운영체제 장애 등으로 네트워크 연결이 차단된 경우	장애 내역을 SMS로 보고함
2급장애	서비스 장애가 발생한 경우	장애 내역을 SMS로 보고함
3급장애	관리자에 의해 주어진 횟수 만큼의 서비스 검사에서 1회 이상 실패가 발생한 경우	장애 내역을 로그에 저장함
정상	서비스 검사가 정상적인 경우	-

3.2 데이터베이스

〈표 2〉은 서비스 감시 대상 시스템에 대한 정보를 등록하기 위한 테이블 구조이다.

표 2. 시스템 등록용 테이블
Table. 2 System Registration Table

필드명	설명
no	일련번호(기본키)
address	시스템 IP 또는 도메인 주소
protocol	감시할 프로토콜(서비스) 지정
period	감시 주기(초단위)
times	2급장애를 판정하기 위한 최대 감시 횟수 (이 횟수 이상 실패시 2급장애로 간주됨)
performance	성능 감시 방법 지정 (snmp=1, wmi=2)

〈표 3〉는 시스템 감시 결과를 저장하기 위한 테이블 구조이다.

표 3. 감시 결과 테이블
Table. 3 Observation Result Table

필드명	설명
sno	감시 시스템의 일련번호(외래키)
date	감시 시각 정보
response	응답 시간(ms 단위)
cpu	CPU 평균 로드 값
memory	물리적 메모리 사용율
disk	하드 디스크 이용율
network	네트워크 카드의 패킷 전송율

3.3 구현

데이터베이스 서비스 및 다양한 네트워크 프로토콜의 서비스 상태를 검사하기 위한 모듈은 Visual Studio C++ 6.0 과 윈속(Winsock) 환경 하에서 구현되었다. 〈표 4〉는 현재 구현된 검사 프로토콜과 그에 대한 검사 방법을 요약한 것이다.

표 4. 감시 프로토콜 및 방법
Table. 4 Observation Protocol and Method

감시 프로토콜	감시 방법
ping	RFC 792 에 따라 감시함
dns	RFC 1034 에 따라 감시함
http, ftp	RFC 2068, RFC 959 에 따라 감시함
smtp, pop3	RRFC 2821, RFC 1939 에 따라 감시함
mssql	sqlping 방식에 의한 데이터베이스 동작 여부 확인 방법
oracle	tnsping 방식에 의한 데이터베이스 동작 여부 확인 방법
odbc	ODBC 방식에 의해 연결된 데이터베이스에 대해 SQL 문장 실행함
tcp, udp	일반 TCP 또는 UDP 포트에 접속한 후 데이터를 송수신하여 감시함

이러한 감시 결과는 응답 시간 형태로 MySQL 데이터베이스에 저장되고 1급/2급장애인 경우에는 SMS 가 전송된다. SMS 전송 모듈은 인터넷 SMS 전송 대행 업체(테크노코리아)[6]의 SMS 전송 프로토콜을 이용하며 인터넷의 단절과 같은 장애가 발생한 경우를 대비해서 곧바로 SMS를 전송할 수 있는 전용장비(KTF Mailshot)[7]를 별도로 구축하여 사용하였다. 관리자를 위한 웹 인터페이스는 윈도우

즈 웹 서버인 IIS(Internet Information Server) 를 사용하였고 서버 스크립트 언어로 PHP 를 사용하였다. 관리자는 감시할 시스템을 웹 인터페이스를 사용하여 MySQL 데이터베이스에 저장하고 감시 모듈은 이러한 설정 정보를 사용하여 감시하는 방식을 취했다. 웹 인터페이스에서 특이할 만한 점은 현재의 검사 결과를 실시간적으로 보고하기 위해 자바(JAVA)의 Live Connect[8] 기술을 사용한 것이다. 이것은 웹 페이지[9]가 자바 애플릿을 사용하여 웹 서버에 접속하고 데이터를 상호 교환할 수 있는 기술을 말하며 여기에서는 감시 모듈에 의한 결과를 실시간으로 웹 페이지에 전송하기 위해 사용되었다.

감시 대상 시스템의 성능 정보를 구하기 위해 네트워크 장비나 유닉스, 리눅스 시스템에 대해서는 SNMP(Simple Network Management Protocol)[10]를 사용하였고 마이크로소프트 윈도우즈 시스템인 경우에는 WMI(Windows Management Instrumentation)[11]를 사용하였다.

Ⅳ. 실험 및 결과 고찰

4.1 실험 환경

라우터를 포함하는 27 대의 노드와 DNS 서버, FTP 서버, 대의 웹 서버, 이메일 서버, 오라클 데이터베이스, MS SQL 데이터베이스, 자체 프로토콜에 의한 서비스 등 총 17 대의 시스템을 감시하도록 구성하였다. 이메일 서버를 제외한 각각의 시스템은 1 분 단위의 서비스 감시를 수행하였고 서비스가 주어진 시간(보통 20초)내에 응답하지 않은 경우 재시도를 수행하도록 하였다. 서비스 요청에 대한 응답이 주어진 시간동안 없는 경우 보통 2 회의 재시도가 수행되도록 하였는데 이러한 재시도는 서비스 응답이 한 번이라도 성공하면 더 이상의 재시도를 하지 않고 3급장애로 간주된다. 만약 3 회에 걸친 감시가 모두 실패한 경우에는 ping 검사를 수행하고 ping 시험 결과가 성공이면 2급 장애로 보고하고 실패한 경우에는 1급장애로 보고한다. 다음 (그림 4)는 라우터에 대한 등록 정보를 보여주고 있다.

이러한 시스템별 구성 환경 정보 외에 연속 장애 발생에 따른 보고 횟수를 제한하는 것 등과 같은 SOS 전반에 걸친 환경 정보가 필요한데 (그림 5)는 이러한 정보를 관리하기 위한 화면이다.

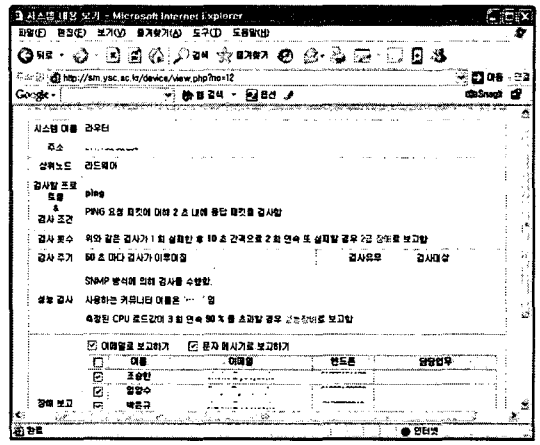

그림 4. 등록된 라우터 정보
Fig 4. Registered Router Details

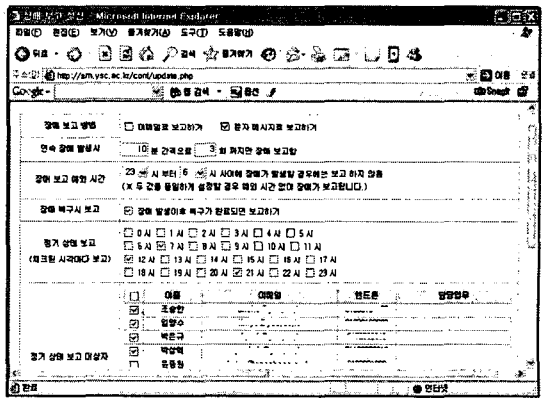

그림 5. 시스템 환경 변수 설정
Fig 5. System Environment Variables Setting

4.2 결과 고찰

시스템에 대한 감시 결과로 응답 시간, CPU 로드 평균, Memory 이용률 등의 정보가 데이터베이스에 저장되는데 이러한 정보를 웹을 통해 조회할 수 있도록 구현하였다. (그림 6)은 하루 동안 측정된 2 분 단위의 평균값을 보여주고 있다. 초록색 막대 선은 응답 시간을 나타내며 선 그래프는 성능 지표를 표시한 것이다.

(그림 7)은 라우터 장비에 대한 것으로 17 시와 18 시 사이에 CPU load 가 90% 이상 치솟는 성능 장애가 발생한 경우이다. 이 경우는 네트워크에 연결된 컴퓨터의 바이러스로 인해 대량 패킷이 발생한 경우로 전체 네트워크 망의 인터넷 장애로 이어질 수 있는 상황을 SOS를 통해 관리자는 즉시 인지하게 되었고 적절한 조치를 취함으로써 전체 망의 장애 기간을 최소화 시킬 수 있었다.

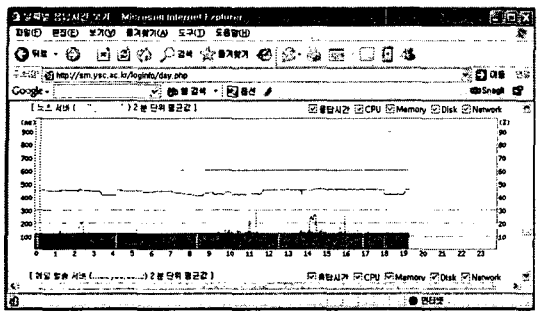

그림 6. 시스템 감시 결과 화면
Fig 6. System Observation Result Window

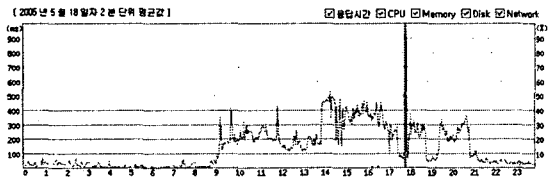

그림 7. 장애가 발생한 경우의 화면
Fig 6. Fault Occurrence Window

V. 결론

본 논문은 서비스의 수행 상태를 감시하기 위한 서비스 감시 시스템을 정의하였고 장애뿐 아니라 성능 초과에 대한 SMS 보고를 수행하며 관리자에게 보다 많은 편의성을 제공하는 웹 기반의 SOS 를 설계 및 구현하였다. 제시된 SOS를 사용하여 관리자는 서비스에 대한 응답 시간뿐 아니라 시스템의 성능 지표인 CPU load, memory usage 등을 그래프 방식으로 쉽게 확인할 수 있게 되었다. 이로써 서비스에 대한 장애뿐 아니라 시스템에 부족한 자원을 쉽게 파악할 수 있게 되었으며 사용자에 의한 장애 보고에 의지하지 않고 장애에 따른 복구 시간을 최소화할 수 있게 되었다. 또한 시스템 성능을 초과하는 서비스에 대해서는 미래에 발생할 수 있는 서비스 지연을 자원의 증설을 통해 사전에 방지할 수 있게 되었다.

참고문헌

[1] TABS laboratories, Server Observer, http://www. tabslab.co.kr/kr/product/serverobserver40/

[2] Woodstone, Servers Alive, http://www.woodstone. nu/salive/

[3] AdventNet, OpManager, http://manageengine.ad ventnet.com/products/opmanager/index.html

[4] Tobias Oetiker, Dave Rand, Multi Router Traffic Grapher, http://www.mrtg.org

[5] 안용학, 박진호, "웹 기반 네트워크 트래픽 모니터링 시스템의 설계 및 구현", 한국컴퓨터정보학회논문지, 6권3호, pp.64-71, 2001

[6] 테크노코리아, SMS 전송서비스, http://www.techno korea.co.kr/product/product_sms1.html

[7] TelQoS, MailShot, http://www.mailshot.co.kr

[8] Vijay Mukhi's Computer Institute, Live Connect http://www.vijaymukhi.com/vmis/lconnect.html

[9] 남태희, "인트라넷과 웹 기반 시스템 설계 및 구현", 한국컴퓨터정보학회논문지, 7권4호, pp.182-187, 2002

[10] William Stallings, SNMP, SNMPv2, SNMPv3, RMON 1 and 2, 3/E, Addison Wesley

[11] Tunsall, Developing WMI Solutions, Addison Wesley

저 자 소 개

조 승 한

2001년 고려대학교 대학원 컴퓨터학과 수료(박사수료)

1994~1998년 삼성전자(주) 전임연구원

1998년~현재 용인송담대학 컴퓨터게임정보과 조교수