통계적 방법을 통한 원격 자원 모니터링 시스템 설계 및 구현

석원홍° 우연옥 강흥식

인제대학교 컴퓨터공학부
babootaeng@naver.com ,poppi99@nate.com, hskang@cs.inje.ac.kr

Design and Implementation of Remote Resources Monitering System through Statistical Method

Wonhong Seok° , YeonOk Woo, HeungSeek Kang
Dept of Computer Engineering, Inje University

요 약

시스템에 문제가 발생했을 때 또는 발생 가능한 사고를 사전에 예방하기 위해서는 여러 가지 방법으로 시스템을 모니터링하고 또한 문제의 원인을 파악하여 문제를 해결하여야 하는 것이 시스템 관리자의 역할이다. 오늘날 여러 복잡한 시스템을 관리함에 있어서 관리자의 입장에서 정보를 보다 편리하고 효율적으로 모니터링 할 수 있는 모니터링 시스템의 필요성이 요구됨에 따라 이런 어려움을 해결하기 위해 다양한 모니터링 시스템들이 개발되어지고 있다. 본 논문에서 제시하는 통계적 방법을 통한 모니터링시스템은 기존의 모니터링 시스템의 단점인 수동적인 모니터링이 아닌 통계치를 활용하여 이상이 있을 시 알려주는 기능을 제안함으로써 시스템을 신속하게 복구하고, 임계값를 이용하여 시스템 이상을 사전에 예방하고자 한다.

1. 서 론

시스템에 문제가 발생했을 때 혹은 발생 가능한 사고를 사전에 예방하기 위해서는 여러 가지 방법으로 시스템을 모니터링하고 또한 문제의 원인을 파악하여 문제를 해결해야 하는데 이런 역할을 수행하는 이를 시스템 관리자라고 한다. 인터넷의 수요 및 전송량의 폭증에 따라, 인터넷환경의 신뢰성 증대 및 성능향상을 위해 시스템 성능 관리가 중요하다. 이로 인해 관리자의 역할 또한 증가하게 되었는데 여러 시스템을 관리하면서 발생되는 복잡하고 번거로움을 관리자의 입장에서 보다 편리하게 모니터링 할 수 있는 모니터링 시스템의 필요성이 요구됨에 따라 다양한 측면에서 모니터링 시스템 개발이 시도되어지고 있다. 또한 시스템의 취약점이나 침해사고가 났을 경우에는 빠른 발견과 그에 따르는 신속한 조치가 필요하다. 본 논문에서는 통계적 방법을 통한 모니터링 시스템을 제안함으로써 시스템을 신속하게 복구하고, 임계값을 이용하여 시스템 이상을 사전에 예방하고자 한다.

이에 본 논문의 구성은 다음과 같다. 먼저 2장에서는 관련연구를 살펴보고, 3장에서 본 논문에서 제안하고 있는 RLMS(Remote Linux system Monitring System)의 설계와 구현 방안을 서술한다. 이어 4장에서는 구현된 RLMS의 실험 결과를 보인 후 마지막으로 5장에서 결론 및 향후 연구방향에 대해서 언급할 것이다.

2. 관련 연구

2.1 기존의 모니터링기법 및 시스템

- 내부 시스템 모니터링 소프트웨어의 활용
  tcpdump, RTSD, logcheck, tripwirre, mrtg, colorlog 등을 활용하여 자신의 시스템을 모니터링 한다.
- 외부 시스템 모니터링 소프트웨어의 활용
  ngrep, snort, NMS, SMS를 활용하여 하나의시스템이 아닌 여러 시스템의 모니터링을 통해 관리자의 효율성을 높여준다. 그러나 현재 시스템들은 대부분 수동적으로 관리대상 시스템의 정보만을 보여 주고 있다.[1] 이로 인해 관리자가 빠르게 대처하지 못하고 있다.

2.2 XBM vs SNMP

기존의 네트워크 및 시스템관리에 가장 많이 사용되고 있는 SNMP를 사용되어져 왔으나 최근 SNMP기반관리에 대한 많은 문제점이 제기 되면서 이 문제를 해결하는 대안으로 XBM(XML-Based Management)이 부각되어지고 있다.[2] 그래서 본 논문에서는 XML를 기반으로 하는 시스템을 구축하고자 한다. XML 기반의 시스템 구축에 있어서 필요한 기술인 파서의 종류인 DOM, SAX와 JAXP의 대해서 서술한다.

2.2.1 DOM과 SAX

DOM의 목적은 광범위한 환경과 응용프로그램에서 표준적인 프로그래밍 인터페이스를 제공하가 위한 것으로 개발자는 XML의 생성, 문서구조의 탐색, 내용의 추가, 삭제을 API인 DOM을 통해 수행할 수 있다. SAX는 XML의 생성,수정 보다는 주로 XML 파싱에 활용되어진다. DOM과 달리 파싱하는 동안에 발생하는 이벤트에 대해서만 반응하고 이를 따로 메모리에 저장하지 않으므로 빠른 수행속도를 가진다.

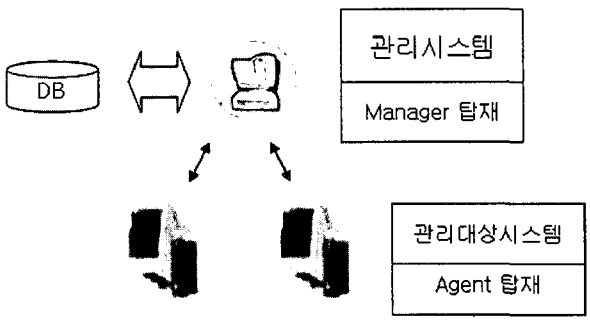
2.2.2 JAXP

본 논문에서는 자바를 이용해서 구현을 하였으므로 자바에서 제공하는 JAXP(Java API for XML Parsing)를 이용한다. JAXP는 SAX와 DOM 파서들을 사용할 수 있도록 인스턴스화 해주는 API이다.

3. 설계 및 구현

3.1 구성도

본 논문에서 제안하고 있는 RLMS의 구성도는[그림 1]과 같다. 시스템은 크게 관리대상 시스템과 관리시스템으로 구성된다. 관리대상 시스템에는 Agent가 탑재된다. Agent에서는 CPU로드 평균, 남아 있는 디스크 공간, 네트워크 트래픽 등과 같은 시스템 성능 관련 수치와, HTTP, SMTP 등과 같은 주요 서비스 사용 가능 여부 등에 대한 정보를 수집하여 XML데이터로 변환하여 관리시스템에 전달하는 역할을 한다. 관리 시스템에 탑재된 Manager에서는 Agent가 수집한 XML데이터를 전송 받아 관리자에게 편리한 GUI환경을 제공하고, 수집된 데이터를 DB에 주기적으로 저장하게 된다. 주기적으로 저장된 데이터를 참고로 관리대상 시스템의 평균적인 시스템 자원상황을 파악하게 된다. 이렇게 축적된 통계적 자료는 현재 자원상황과 비교하여 크게 다를 경우 관리자에게 알려주게 된다.



[그림 1]전체 구성도

3.2 Agent 설계

시스템에서 필요한 정보를 수집하여 XML데이터로 변환하는 역할을 한다. 다음에서 각 정보를 수집하는 방법과 가공하는 방법을 서술한다.

3.2.1 CPU

/proc/stat 에서는 5개의 필드로 구성되어 있다. 첫번째 필드는 CPU번호, 두번째 필드는 user 모드, 세번째 필드는 low priority(nice상태)의 user모드를 나타낸다. 네번째 필드는 system 모드 마지막은 idle 테스크의 jiffies 소모 값이다. 전체 CPU 사용률을 알기 위해선 일정시간 소비된 *idle jiffies*를 총 소비된 *jiffies*로 나누면 원하는 정보를 얻어 낼 수 있다.

3.2.2 메모리

메모리는 크게 물리적인 메모리와 Swap메모리로 구성된다. 필요한 데이터는 할당된 전체 블록크기, 사용되는 블록크기, 사용할 수 있는 블록크기, shared, buffers, chached된 블록크기이다. 이러한 정보는/proc/meminfo에서 제공된다.

3.2.3 네트워크 트래픽

/proc/net/dev에서 데이터 전송량을 얻어 올 수 있다. 100Mbps기준으로 실제 통과하는 패킷누적카운트와 비교 연산함으로써 얻어낸다.

3.2.4 디스크 용량

디스크 용량의 경우는 /etc/mtab와 /proc/mounts를 같이 이용하여 파일시스템 정보를 가져와야 한다. 이렇게 획득된 마운트 장치정보를 이용해서 statfs시스템함수를 이용하여 각 장치의 크기와 사용량을 알 수 있다.

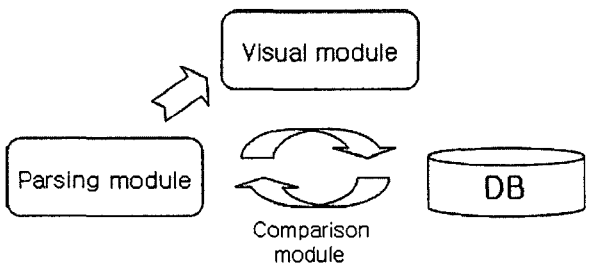
3.2.5 각종 로그파일

로그파일은 /var/log내의 파일들을 파싱 처리하고, XML 데이터 형식으로 재구성하여 Manager에게 전송한다.

3.2.6 기타 각종 정보

관리대상 시스템의 로긴한 유저정보, 프로세스정보 등의 정보를 수집하게 된다.

3.3 Manager 설계



[그림 2] Manager 구조

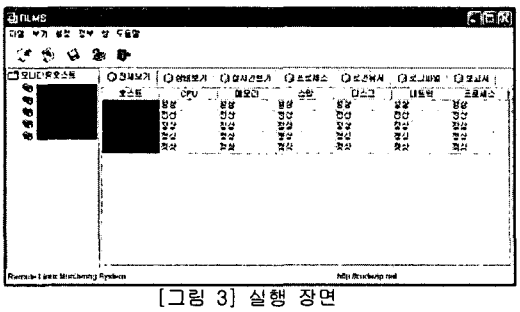
PARSING MODULE

- 관리대상 시스템으로부터 전송받은 정보데이터인 XML데이터를 JAXP API를 이용해서 파싱하는 부분이다. 처리된 데이터는 VISUAL 모듈에서 사용되어지고, DB에 주기적으로 저장되어진다. DB에 저장된 정보는 관리대상시스템별로 통계값을 얻어낸다.

VISUAL MODULE
- 자바의 SWING을 이용해서 파싱된 데이터를 관리자가 손쉽게 알아볼 수 있도록 하는 역할을 한다.

COMPARISON MODULE
- 본 논문에서 제안하는 통계적 방법을 통한 모니터링의 핵심모듈이다. 관리대상 시스템들의 자원 상황을 DB에 시간, 날짜별로 데이터를 저장하게 된다. 시스템별로 저장된 데이터를 통해 통계값을 얻어낼 수 있다. 얻어낸 통계값을 실시간 자원상황과 비교함으로써 시스템 이상을 파악해 낼 수 있다. 또한 관리자가 설정한 임계값 수치이상의 자원상황이 파악될 경우 Visual module에 이벤트를 발생시켜 줌으로써 관리자에게 통보된다.

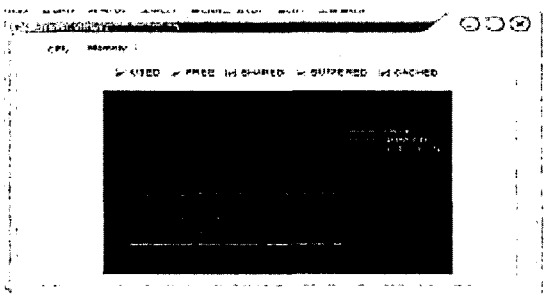
4. 실험 및 결과



[그림 3] 실행 장면

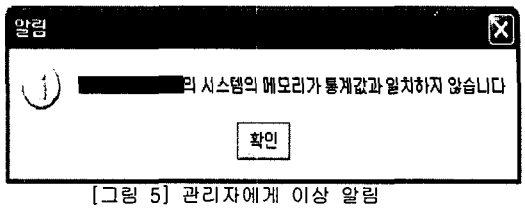
[그림 3]에서 각 관리대상 시스템별로 현재 시스템의 CPU, 메모리, 디스크, 네트워크 등 상태가 정상적으로 동작하는 것을 보여주고 있다. 다음은 RLMS 기능이다.

<전체보기> : 현재 관리대상 시스템들의 자원 상황에 관한 이상 유무를 파악할 수 있다.
<상태보기> : 관리 대상 시스템의 한 클라이언트의 대한 자원 상황 수치를 상세히 볼 수 있다.
<실시간보기> : 관리 대상 시스템의 실시간 모니터링
<프로세스> : 관리대상 시스템의 현재 프로세스 정보들
<로긴유저> : 관리 대상 시스템의 현재 로그인 된 유저들의 정보와 수행중인 작업을 살펴 볼 수 있다.
<로그파일> : 관리대상 시스템의 로그파일을 본다
<보고서> : 시스템의 정보들을 보고서 형식으로 변환



[그림 4] 메모리 실시간 모니터링

[그림 4]는 RLMS의 기능 중에 실시간 자원 상황을 그래픽화하여 보여주는 기능 중 메모리 자원상황을 실시간으로 보여주고 있다. 다음의 [그림 5]는 관리대상 시스템의 메모리 수치가 통계값와 일치하지 않을 경우 관리자에게 통보되는 메시지이다.



[그림 5] 관리자에게 이상 알림

5. 결론 및 향후 연구 방향

본 논문에서는 수동적으로 정보만을 보여줌으로써 관리자가 빠르게 대처할 수 없었던 기존의 모니터링 시스템의 문제점을 보완하고자 통계적 방법을 통한 모니터링 시스템을 제안하였다. 본 시스템은 관리대상 시스템의 자원 상황과 통계값이 일치하지 않을 경우 시스템 문제 이상으로 파악하여 시스템 장애를 즉각적으로 알려 줌으로써 발생장애에 대해서 관리자가 즉시 인지하고 신속하게 복구하도록 유도하였다. 이렇게 함으로써 시스템에 발생된 장애에 의한 피해를 최소화할 수 있다. 모니터링에서 단계적인 임계값을 설정함으로써 장애가 발생하기 전에 장애 가능성이 높은 자원의 사용에 대해 관리자에게 경고함으로써 장애를 예방한다. 향 후 본 RLMS은 시스템에 한정된 모니터링이 아닌 네트워크 장비 및 네트워크 로드간의 상태도 모니터링 할 수 있는 방향으로 발전해 나가야 할 것이다.

6. 참고 문헌

[1] 홍석범, 시스템 및 네트워크 모니터링을 통한 보안강화
[2] 최미정, SNMP와 XML 기반의 네트워크 관리에 대한 비교 및 분석 한국통신학회(2002,4)
[3] Java™ 2 Platform, Standard Edition, V 1.4.3 API Specification (http://java.sun.com)
[4] Harvey M. Deitel. Paul J. Deitel, JAVA HOW TO PROGRAM 5th(2003)
[5] John Wiley,Java, XM, and the Jaxp,Arthur (2002)
[6] http://sax.sourceforge.net
[7] http://www.w3.org/DOM/