

리눅스(Ubuntu)기반 시스템 모니터링 및 자동화 툴 개발 연구

허우솔, 신호섭, 조주필, 김원종, 서종현, 김수미, 최정원, 정가은

국립군산대학교

0202hws@naver.com, hsshin@kunsan.ac.kr, stefano@kunsan.ac.kr, 01055130143@gmail.com, flopssy@naver.com ,
ksm2613@dbit.kr, cjw0803@dbit.kr, jxong_ganny@naver.com

A Study on Developing a System Monitoring and Automation Tool based on Linux (Ubuntu)

Woo Sol Heo, Ho Sub Shin, Jupil Cho, Wonjong Kim, JongHyun Seo, Sumi Kim, Jeongwon Choi, Gaeun Jung
Kunsan Nat'l Univ.

요약

현대 IT환경에서 서버의 안정성은 기업 경쟁력과 직결된다. 그러나 기존의 수동적 관리 방식은 오류 발생 시 신속한 대응이 어렵고 반복적인 작업으로 인한 비효율이 발생한다. 또한 기존 오픈소스 모니터링 시스템은 강력한 기능을 제공하나 초기 설정이 복잡하고 자원 소모가 많아 소규모 환경에 적합하지 않다. 본 연구에서는 리눅스 환경에서 동작하는 경량화된 모니터링 및 자동화 툴을 개발하였다. Python을 활용하여 CPU, 메모리, 디스크, 네트워크 자원을 실시간 모니터링하고, Bash스크립트를 통해 로그 감지 및 서비스 재시작과 같은 자동 복구를 수행하였다. 또한 임계치 초과 시 이메일 알림 기능을 구현하였으며, 선택적으로 Flask 기반 웹 대시보드를 통해 직관적인 시각화를 제공하였다. 실험 결과, 제안된 시스템은 소규모 서버 환경에서 안정적이고 효율적인 모니터링 및 관리가 가능함을 확인하였다.

I. 서 론

현대 기업은 IT 시스템의 안정성과 가용성이 경쟁력과 직결되며, 서버 장애나 성능 저하는 업무 중단과 직결되어 심각한 경제적 손실을 초래할 수 있다. 특히 소규모 기업이나 개인 서버 환경에서는 인력과 예산의 제약으로 인해 전문적인 모니터링 시스템을 구축하기 어렵다.

기준의 서버 관리 방식은 주로 수동 점검에 의존하며, 오류 발생 시 신속한 대응이 어려워 반복적인 작업과 인적 오류가 발생할 가능성이 높다. 상용 모니터링 솔루션은 다양한 기능을 제공하지만, 초기 설정과 유지 관리가 복잡하고 하드웨어 자원을 많이 소모하여 소규모 환경에서는 부담이 된다. 또한 오픈소스 모니터링 시스템(Nagios, Zabbix, Prometheus 등)은 기능이 방대하지만, 경량화 및 간단한 설치를 필요로 하는 소규모 환경에서는 불필요한 기능과 복잡성이 문제로 작용한다.

따라서 본 연구는 리눅스 환경에서 경량화된 시스템 모니터링 및 자동화 툴을 개발하고자 한다. 본 연구의 목적은 소규모 환경에서도 쉽게 적용할 수 있는 도구를 통해 서버 차원을 실시간으로 모니터링하고, 이상 발생 시 자동 알림과 간단한 복구를 수행함으로써 안정성과 운영 효율성을 향상시키는 것이다.[1]

II. 본론

본 연구에서 제안한 경량화 모니터링 및 자동화 툴의 성능을 검증하기 위해 소규모 서버 환경을 가정하여 실험을 수행하였다. 실험 환경으로는 Ubuntu Linux를 사용하였으며, Python 3.8이상과 Bash 스크립트를 기반으로 개발하였다. 시스템 지표 수집을 위해 psutil 라이브러리를 활용하였고, 임계치 알림은 기능은 Python의 smtplib 모듈을, 웹 기반 대시보드는 Flask를 이용하여 구현하였다. 테스트는 가상 서버 환경과 데스크톱 환경에서 진행하였으며 SMTP서버로는 Gmail을 사용하였다.[1][2]



그림 1. 초안 코드 구현(CPU, 메모리, 디스크, 네트워크, 사용량)

위의 페이지에서 필요한 데이터들을 저장하고 관리할 수 있는 코드와 기본적인 구조를 그림 1과 같이 구성하였다. Python과 psutil 라이브러리를 활용하여 CPU, 메모리, 디스크, 네트워크 사용량을 주기적으로 수집하였다. 수집 과정에서 시스템에 부하를 최소화하도록 설계하였으며, 실시간으로 모니터링 정보를 확인할 수 있었다. 이 과정에서 관리자는 시스템 상태를 쉽게 파악할 수 있었으며, 주요 자원의 사용 현황과 트렌드를 직관적으로 이해할 수 있었다.[1]

2. 로그 잡지 및 자동 복구 기능 구현

아래 그림2와 같이 Bash 스크립트를 활용하여 서버 로그에서 오류 메시지를 실시간으로 탐지하였다. 오류가 발생하면 스크립트가 즉시 관련 서비스를 재시작하거나 필요시 프로세스를 종료하도록 설계하였다. 실험을 통해 다양한 오류 상황에서도 툴이 신속하게 복구 작업을 수행함을 확인하였다. 이를 통해 관리자의 개입 없어도 시스템의 연속성을 유지할 수 있음을 입증하였다[1].



그럼 2. 로그감지 및 자동 복구 기능 구현

3. 임계치 기반 알림 기능 구현.

구현된 그림 3을 보면 Python의 smtplib 모듈을 활용하여 임계치 초과 시 이메일 알림 기능을 구현하였다. 설정한 임계치를 초과하는 이벤트 발생시 관리자에게 즉시 알림이 발송되며, 동시에 여러 이벤트가 발생하더라도 모든 알림이 안정적으로 전달되었다. 이를 통해 관리자는 원격에서도 시스템 상태를 실시간으로 파악하고, 필요시 신속하게 대응할 수 있다.[1],[3]

- [1] Python Software Foundation, “Python 3 Documentation,(<https://docs.python.org/3/>)
 - [2] Flask Documentation, “Flask Web Framework(<https://flask.palletsprojects.com/>)
 - [3] E. Choi, S. Park, “A Study on Lightweight Monitoring System for Small-scale Linux Servers,” Journal of Information Technology, vol. 25, no. 3, pp. 101–110, 2021.



그림 3. 임계치 기반 알림 기능 구현

4. 웹 대시보드 모듈 구현.

아래 그림 4와 같이 Flask 기반 웹 대시보드를 구현하여 시스템 상태를 시각적으로 확인할 수 있도록 하였다. CPU, 메모리, 디스크, 네트워크 사용량과 로그 이벤트, 알림 기록 등이 그래프와 표 형태로 직관적으로 표시되며, 실시간 업데이트 기능을 통해 관리자는 서버 상태를 빠르게 점검할 수 있다. 또한 웹 대시보드는 선택적 기능으로, 필요에 따라 사용자 환경에 맞게 활성화할 수 있도록 설계하였다.[2]

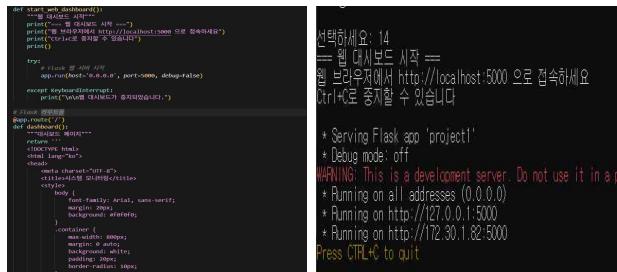


그림 4. 웹 대시보드 모듈 구현.

III. 결론

본 연구에서는 리눅스 환경에서 동작하는 경량화된 시스템 모니터링 및 자동화 툴을 개발하고, 소규모 서버 환경에서의 적용 가능성을 검증하였다. Python과 Bash를 활용하여 CPU, 메모리, 디스크, 네트워크 자원을 실시간으로 모니터링하고, 로그 오류 발생 시 서비스 자동 재시작과 이메일 알림 기능을 수행하도록 설계하였다. 또한 Flask 기반 웹 대시보드를 선택적으로 구현하여, 관리자가 시스템 상태를 직관적으로 확인하고 실시간으로 대응할 수 있는 환경을 제공하였다. 한계점으로는 대규모 서버 환경에서 성능 한계와 고급 분석 기능 미흡, 다중 서버 모니터링 제한이 있다. 향후 연구에서는 머신러닝 기반 이상 탐지 기능, 다중 서버 모니터링, 대시보드 고도화 등을 통해 확장성을 강화할 예정이다.

ACKNOWLEDGMENT

본 논문은 (재)전북테크노파크 재원을 지원받아 수행된 지역특성화산업 전문인력양성사업 연구 결과입니다.

참고문헌