Journal of The Korea Society of Computer and Information
Vol. 24 No. 4, pp. 123-128, April 2019

www.ksci.re.kr
https://doi.org/10.9708/jksci.2019.24.04.123

Implementation of web server monitoring system using crawling technology

Young-Geun Yu*, Ki-Bok Nam*, Koo-Rack Park*

Abstract

In modern society, there are WEB sites that provides information in various fields by many advances in IT technology. As the number of users accessing the Web increases, one server becomes unavailable, and multiple servers are deployed to provide the service. In addition, systems that control servers are required to manage multiple servers. However, server control systems in the existing market are mostly those that notify managers through SMS and apps when a server's down or a controlled port is closed. However, in case of servers that generate a lot of traffic, the Web server and the WAS are operated and managed each independently. The WEB and WAS provide service by connecting to each other. However, the connection between WEB and WAS may be disconnected due to various environmental factors. In this case, the existing control system can not determine whether the service is working properly. Even in the case of WEB and WAS of a server that is operated independently, there is a phenomenon that the existing control system does not know the problem even when the normal service is not provided due to environmental factors such as disconnection to DB. In this paper, we implemented a system to check the normal state of Web service using Web crawling to solve this problem.

▸Keyword: WAS, WEB server, Server Management, Monitoring Systems, Crawling

I. Introduction

최근 급격한 IT기술의 발전으로 서버에 대한 관심이 늘어나고 있다. 또한 다양한 분야의 정보를 제공하는 웹 사이트들이 존재하며 IT분야의 질적 수준 향상을 요구하고 있다. 이에 따라 수많은 기업들은 서버 관련 서비스를 제공하기 위해 사내 시스템 및 고객을 대상으로 서비스 하는 서버의 모니터링 관제 시스템을 운영하고 있다. 이러한 운영시스템들은 서버의 정상 작동을 위한 서비스로 서버나 네트워크 장비를 모니터링 할 수 있는 시스템을 자체적으로 구축하거나 도입하여 운영하고 있다

현재 대부분 운영되고 있는 서버 모니터링 시스템 에서는 자체적으로 정의한 프로토콜에 따라 에이전트와 서버간의 통신이 이루어지고 있으며 서버의 다운 여부와 관제대상 포트가 Close 되었을 때 관리자에게 SMS 또는 앱을 통해 알려주는 시스템이 제공된다. 하지만 다양한 연구개발 실적의 발표에도 불구하고 이미 운용되고 있는 관제 시스템들과 앞으로 개발될 시스템에 대해서는 다음 몇 가지 이유들로 인하여 효율적인 서버 관제 시스템을 기대하기에 다소 부족한 면이 있다. 트래픽이 많이 발생될 경우에는 WEB서버와 WAS를 각각 별도로 관리하며 서로 연결되어 서비스가 된다. 또한 단독으로 운영되는 서버의 WEB, WAS인 경우에도 DB 접속 단절 등 여러 환경적인 요인으로 인해 단절될 경우 기존 관제 시스템을 통해서는 서비스의 정상작동 여부를 판단하지 못한다[1,2]. 또한 단독으로 운영되는 서버의 WEB 및 WAS의 경우 DB 연결 단절 등의 환경적인 요인으로 정상적인 서비스가 되지 않는 경우에도 기존의 관제 시스템은 문제점을 알지 못하는 현상이 있다. 본 논문에서는 이러한 문제를 해결하기 위해 웹 크롤링을 이용하여 WEB 서비스의 정상적인 상태를 확인하는 시스템을 구현한다.

• First Author: Young-Geun Yu, Corresponding Author: Koo-Rack Park
*Young-Geun Yu (ygyu@krict.re.kr), Dept of Computer Science & Engineering. Kongju National University
*Ki-Bok Nam (mtgood@naver.com), Dept of Computer Science & Engineering. Kongju National University
*Koo-Rack Park (ecgrpark@kongju.ac.kr), Dept of Computer Science & Engineering. Kongju National University
• Received: 2018. 09. 04, Revised: 2018. 10. 22, Accepted: 2019. 04. 29.

본 논문은 다음과 같이 구성한다. 2장은 현재 사용되고 있는 모니터링 알람시스템, 그리고 웹 크롤링과 HTML DOM, JavaScript에 대하여 기술한다. 3장에서는 본 논문에서 제안한 효과적인 모니터링을 위한 알람 시스템의 설계, 구성에 대하여 기술하고, 4장에서는 구현된 서버 모니터링 시스템을 시연하였다. 마지막으로 5장에서는 결론과 본 논문과 연관된 향후 연구에 대하여 기술한다.

II. Preliminaries

1. Related works

1.1 RTMS

LiveMedia에서 쉬운 서버관리를 위해 구축하는 모니터링 시스템인 RTMS(Real-Time Total Monitoring System)는 다수의 PC나 서버의 장애요소와 세부현황을 실시간으로 감지하여 자동문자 서비스와 모바일 뷰어를 통해 장소 상관없이 쉽고 빠르게 확인할 수 있는 기술로 장애 발생 후 장애 알람 문자를 전송하여 PC나 모바일을 통해 언제 어디서든 서버상태와 장애를 파악할 수 있다. 서버장애와 서비스장애, 네트워크에 장애가 발생하면 초단위로 실시간 감지하여 자동문자 양식 및 사용자정의 메시지를 전송하고 PC, 모바일을 통해 언제 어디서든 빠르게 서버상태 및 장애를 파악하며 최초 장애 발생 후 PC, 모바일을 통해 원격접속 및 제어를 할 수 있다. 별도의 프로그램을 서버나 장비에 설치하지 않아 시스템과 네트워크에 아무런 영향을 주지 않는 안정성과 한 대의 모니터링 서버로 여러 시스템과 네트워크를 지원하여 모니터링 구축비용을 절감할 수 있는 절약성, 감시대상인 클라이언트 서버 추가 시 손쉽게 등록 할 수 있는 확장성, 시스템 관련 문서의 통합문서 관리기능으로 일괄 관리가 가능하게 하고 모바일과 접근권한 설정을 지원하는 일괄성, 이 네 가지 특징을 가지고 있다. 서버의 성능이나 현황을 종합하여 영구적인 로그 및 DB를 통해 장기간의 시스템 운영에 필요한 성능분석 자료와 정확한 통계자료를 제공하여 시스템의 안정성을 높여주고 시스템 관련 문서와 자료를 통합 DB화하여 시스템운영의 효율과 일괄관리를 기대할 수 있다. 시스템 관리비용은 줄이고 업무효율을 높인 실시간 종합 모니터링 시스템이다[3]

1.2 Crawling

크롤링은 웹 크롤러에서 출발한 말로 인터넷상의 페이지(HTML, 문서 등)를 수집해서 분류하고 저장하며 쉽게 찾아볼 수 있는 역할을 하는 일종의 로봇이다. 웹 크롤러의 행위는 복합적 정책들의 산물이다. 크롤링 정책에는 페이지의 다운로드를 언급하는 선택정책과 문서의 변경사항을 언제 검사할지 언급하는 재 방문 정책, 이 정책은 변경 빈도에 상관없이 동일한 빈도로 컬렉션의 모든 페이지를 다시 방문하는 통일 정책과 자주 변경되는 페이지를 더 자주 방문하는 것으로 포함하는 비례정책 이 두 가지 정책으로 나누어진다. 두 경우 모두 반복되는 페이지의 크롤링 순서를 임의 순서 또는 고정 순서로 수행 할 수 있다. 그리고 웹사이트의 과부화를 막기 위한 언급하는 공손성 정책과 분산 웹 크롤러를 어떻게 조율할지 언급하는 병렬화 정책이 있다. 웹 크롤러는 대체로 모든 페이지의 복사본을 생성하며, HTML 코드 검증과 링크를 주기적으로 체크하며 웹 사이트의 자동 유지 관리 작업 등 웹 페이지의 특정 형태의 정보를 수집하는 데도 사용된다. 인터넷에 공개되어 있는 웹페이지의 HTML을 수집하기 위해 사용한다. 웹사이트 데이터를 크롤링하기 위해서는 웹사이트의 구조를 파악해야한다[14, 15]. 즉 브라우저에 URL입력, 링크를 누를 때 웹 애플리케이션은 데이터베이스로부터 요청 받은 자료를 찾아 웹 페이지에 해당 자료를 삽입한다. 따라서 일정한 패턴을 가진 HTML 템플릿으로 기술된 웹 사이트에서 웹페이지를 크롤링하기 위해서는 템플릿의 어떤 부분에 삽입된 데이터를 추출할지 결정해야 한다[4,5].

웹페이지를 자동으로 수집하기 위해서는 우선적으로 웹페이지의 URL생성 패턴을 파악해야하며 파싱할 때 인터넷에서 URL주소를 통하여 접근할 수 있는 웹페이지는 HTML 언어로 작성되어 있으므로 HTML문법을 이해하여 HTML 코드에서 원하는 데이터를 수집한다. 따라서 XML이나 HTML과 같은 특정한 규칙에 근거하여 생성된 데이터를 분해하는 파싱 과정을 거쳐 사용자가 지정한 저장규칙에 따라 데이터를 분해하여 저장한다[13].

1.3 DOM

Html문서에 접근하기 위한 표준 객체 모델이자, 프로그래밍 인터페이스인 DOM(Document Object Model)은 문서 객체 모델이라고도 부르기도 한다. HTML 태그를 자바스크립트에서 사용할 수 있는 객체로 만든 것을 문서객체라고 하고 속성, 메서드, 이벤트, 컬렉션 및 데이터의 명명 규칙을 따르며, 모든 이름은 단일 문자열을 형성하기 위해 함께 연결된 하나 이상의 영어 단어로 정의된다. 브라우저는 웹문서(HTML, SML, SVG)를 로드한 후, 파싱하여 해당 요소(element)들을 객체 모델로 표현하는데 이것이 DOM 이다[6]. 이 DOM은 자바스크립트를 통해 동적으로 변경할 수 있으며 변경된 DOM은 랜더링에 반영된다. HTML과 CSS는 tag, attribute 등을 미리 정해 놓은 뒤 요소를 배치하는 언어이며 HTML은 중첩관계는 대체로 부자관계를 갖는다. CSS는 캐스케이딩, 상속으로 이뤄져있다. 이 요소들을 객체화해서 만든 것이 DOM이다[11,12].

1.4 JavaScript

JavaScript는 넷스케이프사에서 개발 한 객체 지향 언어이다. 원래 명칭은 LiveScript로 개발당시 HMTL 문서에 삽입되어 동적 효과를 제공하기 위해 사용되었다. JavaScript가 삽입된 HTML 문서는 넷스케이프사의 Navigator 브라우저에서 볼 수 있었고, 네비게이터 브라우저는 JavaScript 소스 코드 프로그램을 실행하는 해석기를 가지고 있었다. 초기에 Web 환경에서 HTML 문서에 사용 된 JavaScript 언어는 Web의 사용이 증가함에 따라 JavaScript에 점점 익숙한 개발자가 늘고 있다

[7]. 또한 웹에서 사용되는 기술이 운영체제나 사용 환경을 위한 스크립트 언어로도 사용된다. 또한 인터프리터 소스 코드 프로그램을 해석하고 실행하기 위해 JavaScript는 플랫폼에 의존하지 않는다. 이 점이 여러 환경에서 JavaScript를 사용할 수 있는 장점 중 하나가 된다. 자바와 비슷한 문법을 사용하지만, 자바와 달리 프로토타입 기반 객체 시스템이며 객체는 속성 집합이다. 클라이언트 JavaScript는 웹 브라우저에 내장된 인터프리터에 의해 실행되며 문서객체 모델에 결합되어 작동한다. 웹 페이지는 JavaScript로 작동된 스크립트를 포함 할 수 있는, 이 스크립트는 DOM을 사용하여 웹 페이지를 수정하거나 웹 페이지를 보여주는 웹 브라우저를 제어할 수 있다[8,9]. 최근 들어, JavaScript는 간단한 Web 페이지의 사용자 상호 작용을 넘어, 웹 어플리케이션 (이하, 웹 앱)의 복잡하고 다양한 동작을 수행 할 목적으로 확장되어 사용되고 있다. 웹 응용 프로그램은 다양한 기능의 애플리케이션을 브라우저에서 실행할 수 있도록 구현 된 앱이며 HTML, CSS, JavaScript 기반의 Web 언어로 작성되어 이식성이 좋고, 개발자들에게 많은 인기가 있다[10].

III. The Proposed Scheme

본 논문은 웹 크롤링 기술을 이용하여 웹 서비스의 장애를 빠르게 파악하고 관리자에게 알려주는 시스템을 구성하였다.

다음의[Fig.1]은 제안 시스템의 실행 순서이다.

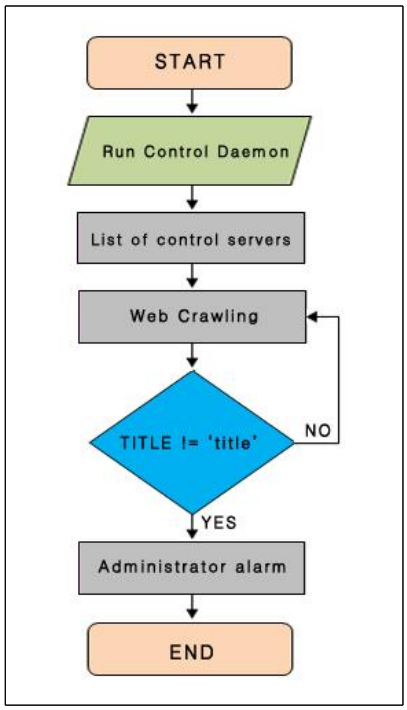

Fig. 1. System sequence diagram

첫 번째, 서버 관제 프로그램을 실행한다.

두 번째, excel 파일로 구성된 관제 대상 서버의 목록 및 정보를 가져온다.

세 번째, 관제 서버의 URL정보로 웹 크롤링을 실시한다.

네 번째, 크롤링한 데이터 중 "TITLE" tag의 정보 내용을 가져와서 관제 서버의 정보와 동일한지 체크한다. 다섯 번째, 크롤링한 데이터 조건에 따라 문제발생이 되면 관리자에게 알람 메시지를 전송한다.

다음[Fig.2]는 관제 서버의 서비스 상태를 확인하기 위한 정보를 저장한 excel 파일의 구성 화면이다

	A	B	C
1	체크URL	타이틀명	연락받을 전화번호
2	https://nid.naver.com/nidlogin.login	네이버 : 로그인	010-0000-1111

Fig. 2. Excel file configuration

A2는 excel 구성 중 체크 URL에 해당하는 관제해야 할 서버의 접속 URL이다.

B2는 A2의 URL에 정상접속 되면 나와야 하는 타이틀 명이다.

C2는 B2의 타이틀 명과 맞지 않을 경우 관리자에게 SMS를 보내기 위한 전화번호이다.

이와 같은 파일구조로 손쉽게 모니터링해야 할 서버를 추가 또는 내용 수정을 할 수 있도록 구성 하였다.

다음 [Fig.3]는 excel 파일로 구성된 관제 대상서버의 목록을 가져오는 소스의 일부이다.

```
fis = READ(엑셀파일경로)
workbook = READ(fis)
sheet = READ(workbook)
FOR 행의 수 만큼 반복
{
  FOR 칼럼 수 만큼 반복
   {
      IF 셀값 != NULL THEN
      {
               switch 셀타입
              {
                SET 타입별 데이터 대입
              }

               IF 첫번째 칼럼 THEN
                 대상사이트 = 칼럼값

               IF 두번째 칼럼 THEN
                  제목 = 칼럼값

               IF 세번째 칼럼 THEN
                  핸드폰번호 = 칼럼값
      }
   }
}
```

Fig. 3. Excel file load

첫 번째, "fis=READ(엑셀파일경로)"에서 엑셀 파일을 Open 한다.

두 번째, "workbook = READ(fis)"는 엑셀의 workbook를 읽어온다.

세 번째, "sheet = READ(workbook)"는 엑셀 Sheet 중 첫 번째를 선택한다.

네 번째, "FOR"문을 이용하여 행의 수만큼 반복 하고 서브 "FOR"문을 이용하여 칼럼수 만큼 반복 실행 한다.

다섯 번째, value값이 숫자 인지 문자 인지 타입별로 데이터를 가져온다.

여섯 번째, 해당 칼럼에 value값을 가지고 와서 처리하고자 하는 변수에 대입 시킨다.

다음[Fig.4]는 서버를 관제하기 위한 웹 크롤링 소스의 일부분이다.

```
WebClient driver = new
WebClient(BrowserVersion.FIREFOX_3);
Page driver_sebu = driver.getPage(site);
 if( !((HtmlPage) driver_sebu).getTitleText().equals(title) )
 {
    Send_Sms(site,phones);
 }
```

Fig. 4. Web crawling source

첫 번째, 크롤링 소스는 java로 구현하였으며 크롤링 엔진은 HtmlUnit 2.31 API 버전을 이용하였고 WebClient를 이용하여 브라우저 버전을 파이어폭스로 설정 하였다.

두 번째, HTML DOC를 가져오기 위해 "driver.getPage()" 함수를 이용하였다.

세 번째, TITLE 정보를 가져오기 위해 "getTitleText()" 함수를 이용하였고 가져온 HTML DOC에서 취득한 TITLE 정보와 관제 서버의 설정된 title 정보가 같은지 체크를 한다.

네 번째, TITLE정보가 서로 다를 경우는 관제서버의 문제가 발생된 것이기 때문에 "Send_sms()"함수를 이용하여 지정된 관리자에게 문자를 발송한다.

IV. Experiments and discussion

다음 [Fig.5]는 기존 관제 시스템에서는 정상관제가 되지 않는 사이트 이다.

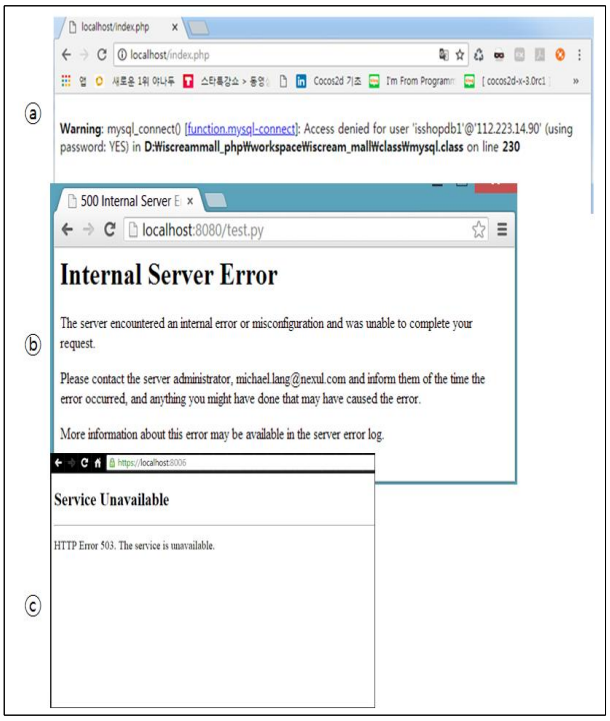

Fig. 5. Controlled site

a형은 DB쪽 서버의 문제 등으로 인한 DB 접속 오류가 발생했을 경우이다.

b형은 여러 가지 상황이 있을 수 있지만 대부분 NAS등 별도 외부 디렉토리와 연결과정에서의 장애로 인한 오류이다.

c형은 서버의 접속자 수가 많거나 접속은 하였으나 타임아웃이 발생 될 경우 또는 서버 내 애플리케이션의 문제로 웹상에서 요청한 서비스가 정상 처리되지 못한 경우에 발생하는 오류 이다.

위 내용처럼 여러 가지 오류가 발생하여도 포트를 모니터링 하는 기존 관제 시스템에서는 이런 오류를 관리자에게 알려줄 수 없어 서비스 장애가 지속되는 경우가 있다.

다음[FIg.6]은 본 논문에서 구현한 웹 크롤링을 이용하여 웹서버의 정상 작동 여부를 판단하기 위한 데몬 프로그램 실행 화면이다.

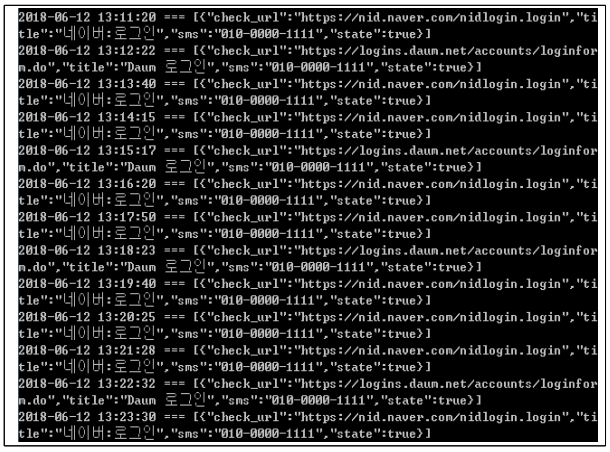

Fig. 6. Run the daemon program

Java로 구현한 웹 크롤러로 1분단위로 웹 크롤링하여 결과를 보여주는 화면이며 테스트를 위해 네이버 및 다음을 웹 크롤링하여 진행 하였다. “check_url” 에서는 접속한 url를 나타냈으며 “title”에서는 웹 서버의 "TITLE TAG"에 해당하는 문자를 비교하기 위한 정보이다. 또한 문제가 발생 되었을 경우에는 title이 서로 다르기 때문에 "SMS"에서 정의한 핸드폰으로 문자를 발송하도록 설계 하였다. 마지막으로 “state"에서는 결과 상태 값으로 정상 접속 시에는 ”true" 문제 발생 시 에는 ”false”로 표기하도록 하였다.

다음[Fig.7]은 웹 사이트의 정상 접속여부를 판단하기 위한 화면이다.

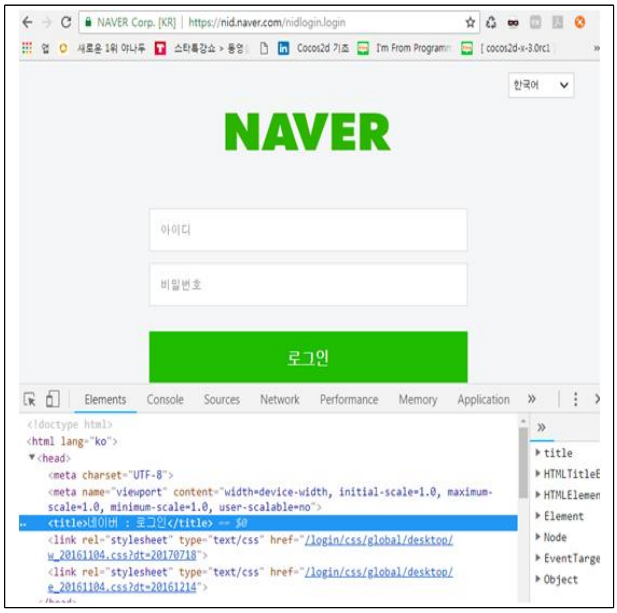

Fig. 7. Title Tag check

해당화면은 네이버의 로그인 화면으로 웹 브라우저의 개발자도구(F12)를 통해 웹 사이트의 HTML 소스를 확인 할 수 있다. 해당 HTML DOM에서 “TITLE TAG"의 TEXT 내용을 확인 할 수 있다. 확인된 title은 본 논문에서 정의된 excel 파일에 기입하여 사이트 정상접속 여부를 확인할 때 이용한다.

다음[Fig.8]은 웹 사이트의 장애가 발생 되었을 때 관리자에게 문자가 발송되는 화면이다.

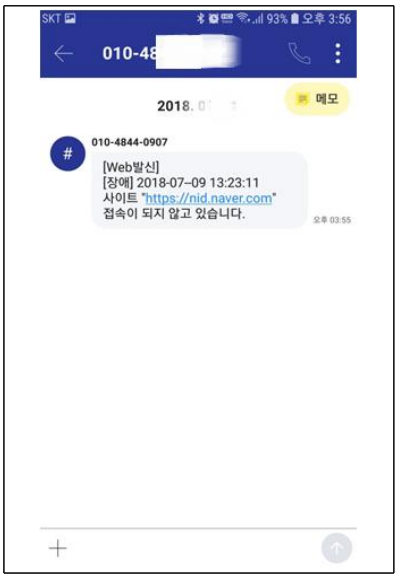

Fig. 8. Fault notification screen

장애가 발생 되었다고 가정하기 위해 [Fig. 2]에 있는 excel 파일의 타이틀 내용을 “네이버 : 로그인”에서 “장애 TEST”로 변경 하여 테스트를 진행하였으며 TEST 후 바로 장애가 발생하였다고 SMS가 발송되었다.

V. Conclusions

IT산업의 발전으로 고객들에게 많은 정보를 제공하는 웹 사이트가 많이 늘어나고 있다. 또한 웹 서비스가 이용자가 증가할수록 서버의 수가 늘어나 관제 대상 서버는 많아지는데 기존 관제 시스템에서는 단순 PORT활성화를 대상으로 모니터링을 하고 있어 환경적인 요인으로 웹 서버의 정상적인 서비스가 이루어지고 있지 않은 경우 기존 서버 관제 시스템으로는 모니터링 하기 어려움이 있다. PORT는 활성화 되어 있으나 DB연결에 문제가 발생했을 경우 또는 사용자가 급증하여 웹에 동시 접속가능한 수를 넘었을 경우등 이런 예측하지 못하는 상황이 발생 했을 경우는 서버의 정상작동 여부를 체크 못 해 본 논문에서 웹 크롤링 기술을 이용하여 HTML TAG중 “TITLE“의 내용을 비교하여 실제 사이트를 이용하는 이용자가 보는 웹 사이트를 모니터링 하고 문제가 발생 될 경우 관리자에게 신속하게 문제를 알려주기 위해 SMS를 발송 할 수 있는 시스템을 구현하였다.

향후 연구에서는 이용자 수 증가에 따른 사이트 접속 속도 저하의 원인 및 문제점을 찾아주는 모니터링 시스템을 연구 하겠다.

REFERENCES

[1] Kyungryun C, Donghan S, SeongJung K, Sumin H, Soochan H. "Multi Visualizing Method for Datacenter Server Monitoring" Proceedings of 1st IITA Korea Information Science Society 2017 Korea Computer Engineering Conference, pp. 268-270, 2017.

[2] Yucheng,L., L. Yubin., "A Monitoring System Design Program Based on B/S Mode" 2010 International Conference on Intelligent Computation Technology and Automation, pp. 184-187, May 2010.

[3] RTMS Project, http://livemedia-soft.com/

[4] S. Ye, J. Lang, and F. Wu. "Crawling Online Social Graphs", In Proceedings of the 12th International Asia-Pacific Web Conference, pp. 236-242. IEEE, 2010.

[5] C Seung-ju, K Jongbae. "Examine the Relationships Between Portal Article of Naver and Real Time Search Word Using Web Crawling", In Proceedings of Society for Humanities and Social Sciences Convergence, pages 787-794. KCI, 2017.

[6] S. H. Jensen, M. Madsen, and A. Møller. "Modeling the HTML DOM and browser API in static analysis of JavaScript web applications", In Proceedings of the Symposium on the Foundations of Software Engineering, September 2011.

[7] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March, 2014.

[8] S.-W. Lee, and S.-M. Moon, “Selective Just-in-Time Compilation for Client-side Mobile JavaScript Engine,” International Conference on Compilers, Architectures and Synthesis of Embedded Systems, ACM, pp. 5-14, 2011.

[9] J. Oh, and S. -M. Moon, “Snapshot-based Loading-Time Acceleration for Web Applications,” Proc. Of the 13th Annual IEEE/ACM International Symposium on Code Generation and Optimization, IEEE, pp. 179-189, 2015

[10] H. Park, M. Cha, and S. -M. Moon, "Concurrent JavaScript parsing for faster loading of Web apps," ACM Transactions on Architecture and Code Optimization, Vol. 13, Issue 4, No. 41, 2016

[11] W3C,"DocumentObjectModel(DOM)",http://www.w3.org/DOM/,2009

[12] W3C,"DocumentObjectModel(DOM)Level1Specification"http://www.w3.org/TR/REC-DOM-Level-1/,1998

[13] "interpreter and JavaScript Engine", http://huns.me/development/360, 2013.10.

[14] "Robots Darabase", http://www.robotstxt.org/db.html

[15] "Web crawler" Wikipedia, http://en.wikipedia.org/wiki/Web_crawler

Authors

Young-Geun Yu received the M.S. and is proceeding Ph. D degree in Computer Engineering from Kongju National University, Korea, in 2016 respectively He is interested in Information security.

Ki-Bok Nam received the M.S. and is proceeding Ph. D degree in Computer Engineering from Kongju National University, Korea, in 2017 respectively He is currently interested in AI, DRM and Web Crawling, and IoT.

Koo-Rack Park received the B.S. degree in Electrical engineering from chung-ang university, Korea in 1986. the M. S. degree in Computer science from soongsill university, in 1988. and Ph. D degree in Science Compute from kyonggi University in 2000. Dr. Park joined the faculty of the Department of Computer Science & Engineering at Kongju National university, Kongju, Korea, in 1991. he is currently a Professor in the Department of Computer Science & Engineering, Kongju National University. He is interested in Information and Communication and Management information and e-commerce