

AJAX+XML 기반의 모니터링 시스템

최 윤 정* · 박 승 수**

Realtime Monitoring System using AJAX + XML

Choi, Yun Jeong · Park, Seung Soo

〈Abstract〉

Nowadays, according to rapid development of computing environments, information processing and analysis system are very interesting research area. As a viewpoint of data preparation-processing-analysis in knowledge technology, the goal of automated information system is to satisfy high reliability and confidence and to minimize of human-administrator intervention. In addition, we expect the system which can deal with problem and abnormal error effectively as a fault detection and fault tolerance. In this paper, we design a monitoring system as follows. A productive monitoring information from various systems has unstructured forms and characteristics and crawls informative data by conditions and gathering rules. For representing of monitoring information which requested by administrator, running-status can be able to check dynamically and systematic like connection/closed status in real-time. Our proposed system can easily correct and processing for monitoring information from various type of server and support to make objective judgement and analysis of administrator under operative target of information system. We implement semi-realtime monitoring system using AJAX technology for dynamic browsing of web information and information processing using XML and XPATH. We apply our system to SMS server for checking running status and the system shows that has high utility and reliability.

Key Words : Monitoring System, Semi-Realtime System, Fault Tolerance, Error Correction, AJAX, XPath, XQuery

I. 서론

최근의 정보환경이 급속도로 발전함에 따라 사용자의

요구에 실시간으로 응답 가능한 정보검색과 더불어 빠르고 정확한 정보처리와 분석시스템에 대한 관심이 높다. 정보의 수집-처리-분석-피드백의 반복 과정으로 수행되는 지식공학의 관점에서는 일련의 각 단계들이 잘 동작하는지 관찰하는 관리자의 개입시기와 역할을 최소화키

* 서일대학 정보통신과(교신저자)

** 이화여자대학교 컴퓨터공학과

면서도 높은 신뢰도를 가지고 원활히 수행될 수 있는 자동화 시스템에 대한 요구가 높다. 더 나아가 관리자의 부재 시 시스템에 문제나 결함이 발생하였을 때 전체적인 시스템의 목표업무나 성능에 적절하게 대처할 수 있는 오류감지(fault detection) 및 정정기능(fault tolerance)이 탑재되어 있기를 기대하고 있다.

정보의 수집과 처리를 담당하는 시스템은 정보 검색기술과 데이터베이스 기술의 발달로 인해 이미 양적으로 많은 성장이 이루어져 있다. 반면, 시스템이 정확히 동작하고 안정되게 수행되고 있는지 확인하고, 자동으로 상태 정보(status information)를 관찰할 수 있으며 시스템과 이상 징후에 대한 적절한 보고(report) 체계가 확립된 질적인 성장이 더욱 절실히 필요하다. 최근의 정보시스템의 운영형태는 과거 소수의 워크스테이션 컴퓨터에서 대규모의 서버 컴퓨터로 옮겨가고 있으며, 독립적인 단위에서 서로 물리적/논리적으로 연결된 상태로 운영되는 구조로 확장되고 있다. 이러한 유기적인 시스템의 정보를 처리하고 있는 동작상태의 정보와 특정 시도에 대한 성공 및 오류결과를 감사하고 보고하는 모니터링의 중요성은 그 규모를 떠나 계속적으로 강조되어야 하는 중요한 요소이다[1-3].

이러한 중요성의 변화를 이끈 원인으로는 정보환경의 발전과 함께 확장된 기업의 정보시스템과 해당시스템의 운영목표에 의해 개발된 시스템의 구조, 그리고 운영방법에 있다. 운영 목적의 다양성은 시스템의 설계 구조에 반영되고, 구조의 다양성은 수집대상이 되는 정보의 형태와 수집 후 처리되어야 하는 형식의 다양성을 만족시켜야 한다. 제조업자가 제공하는 전통적으로 규격화 되어있던 구조대로 관리되는 데이터와는 근본적인 차이가 있는 부분이다. 정보시스템의 변화에 따라 모니터링의 목표 및 정보도 변화되고 데이터의 형식과 내용, 그에 대한 표현방식, 분석을 위한 판단기준이 필요하게 된 것이다. 이러한 상황을 쉽게 수용하여 유연하게 동작하도록 하고 문제에 강건하게 대처할 수 있는 종합적인 모니터링 시스템을 설계하는 것이 필요하다.

본 논문에서 설계하고자 하는 시스템의 배경 및 목표는 다음과 같다.

- 여러 이종의 시스템에서 생성되는 모니터링 정보는 다양한 형태와 특성을 가진다.
- 이러한 정보들은 이질성이 크고 각기 다른 표현양식으로 저장되어 있으므로 일련의 정해진 조건이나 규칙에 따라 수집해야한다.
- 모니터링 정보들을 최종 관리자가 효율적으로 분석할 수 있도록 나타내기 위해서 접속한 서버의 종류, 접속한 위치에 관계없이 관리자가 선택한 탐색 조건과 규칙에 따라 처리되어야 한다.
- 유동적이고 유기적인 관계로 연결되어 있는 서버들의 동작 상태가 실시간으로 확인되어야 한다.

본 논문에서는 정보시스템의 운영목표 하에 관리자의 용이한 분석이 가능하도록 적절한 형태로 변환하여 제공하고 대량의 정보처리에 드는 시간지연을 최소화한 준실시간 처리를 위하여 XML기반 기술을 활용한다. 이에 XML과 XPATH를 이용한 정보처리기술과 함께 웹 정보의 원활한 브라우징을 돕는 AJAX 기술을 이용하여 준실시간 모니터링 시스템을 구현하도록 한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 모니터링 시스템이 웹 환경에서 동적으로 표현되고 준실시간으로 관리되도록 하는 기법에 대해 살피고, 3장에서 본 논문에서 제안하는 모니터링 시스템의 구조와 최적화방안에 대해 소개한다. 4장에서는 본 논문에서 설계한 시스템의 국내 금융 사이트에 적용하여 활용하는 예를 보이고 5장에서 결론을 내린다.

II. 관련 연구

2.1 유비쿼터스 환경에서의 모니터링 시스템

기존의 일반적인 모니터링 시스템은 서버와 어플리케이션의 자원, 네트워크 상황 정보 등을 수집한다. 단위

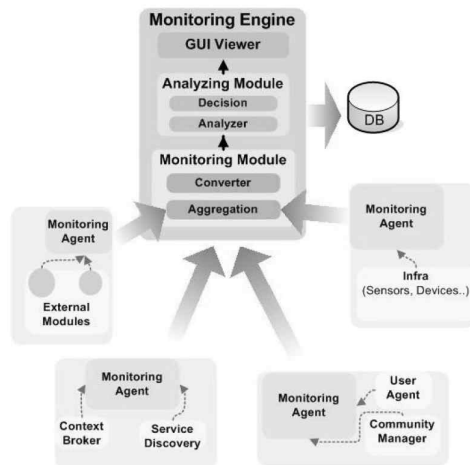
어플리케이션의 실행 상태의 확인을 위해서 운영환경과 개발과, 평가에 걸친 여러 영역의 부분들을 모니터링 하고 있다[1, 3]. 그러나 확장된 응용환경과 다양한 디바이스들에서 수행되는 여러 어플리케이션이 복합적으로 연결되어 하나의 서비스로 제공되는 유비쿼터스 환경에서는 이러한 다양한 서비스들이 지속적이고 정확하게 제공되어야 한다. 이를 위해 각각의 이질적인 어플리케이션으로 부티의 정보수집과 동적인 응답처리, 신뢰성을 검사해야 하고 무엇보다 여러 디바이스들로부터의 상황정보들이 실시간으로 모니터링 되어야하는 점이 중요한 요소로 강조되고 있다[2, 5].

기존 분산 응용환경에서의 계층적 모니터링 기법을 강조했던 시스템[4]은 다양한 자원과 이질적인 통신 프로토콜을 고려하여 컴포넌트 기반의 시스템을 구성하고 다양한 통신 프로토콜을 지원하여 모니터링 할 수 있는 방법이 제안되었다. 이 시스템에서는 SQL 쿼리를 이용하여 자원과 상황에 대한 정보를 수집한다. 따라서 모든 서비스와 어플리케이션의 실행정보는 데이터베이스로 저장되어야 하고 쿼리에 의해 모니터링 정보가 수집되므로 실시간 변화되는 환경변수의 상황을 감시하기 어려운 점이 있으며 쿼리 수행에 따른 성능저하 및 정보의 재구조화에 대한 작업이 뒤따른다. 어플리케이션의 성능을 개선하기 위한 연구 중 하나인 다중-소스 성능분석[1]에서는 어플리케이션의 코드분석 데이터, 서버 및 네트워크 자원, OS 성능 분석 데이터를 수집하고 있다. SQL 서버에 데이터베이스 형태로 저장된 데이터는 분석 후 어플리케이션의 소스 레벨에서 성능 개선에 사용된다.

이러한 기존의 모니터링은 서버 자원 정보를 모니터링하거나 단위 어플리케이션보다 비즈니스 차원에서 통합된 차원의 모니터링이 수행되어 단위 별 수행에 따른 자원의 모니터링에 취약하다는 특성이 지적된다. 또한 모니터링 시스템이 소스 결합을 통해 어플리케이션 단위의 모니터링을 수행함으로써 실시간 실행상황 정보를 얻기 힘들다는 단점이 있다.

유비쿼터스 시스템을 위한 실시간 모니터링 에이전트

[2]는 분산 응용환경에서 여러 서비스들이 수행되는 환경에서의 서비스 모니터링을 위한 연구로서, 서비스의 정확성을 평가하고 신뢰성을 검사하며 피드백 할 수 있는 실시간 모니터링 모델을 제안하고 있다. <그림 1>은 다양한 환경하에서 동작되는 어플리케이션 및 서비스들의 상태와 에이전트의 특성을 고려하여 구성한 모니터링 시스템의 구조도를 나타낸다. 모니터링 엔진은 각 서비스가 수행되는 컴퓨터의 모니터링 에이전트로부터 서비스 모듈의 상황정보를 받아 모니터링 서버에게 전달한다. 분석 모듈은 이 데이터들로 서비스 모듈의 상황을 분석하고 판단한다.



<그림 1> 유비쿼터스 환경을 고려한 모니터링 시스템의 구조도

이 연구에서는 데이터의 수집 및 전달을 위해 SQL 쿼리를 사용하는 대신 JMX(Java Management eXtension) 기능을 사용하고 있다[8]. 또한 JDI(Java Debug Interface)를 이용하여 메서드를 분석한 후, JNI(Java Native Interface)로 서버의 자원정보를 얻고 있으므로, 분산 환경에서 동작하는 유비쿼터스 모니터링 시스템이 보다 안정적이고 높은 응답성을 갖도록 설계되었다.

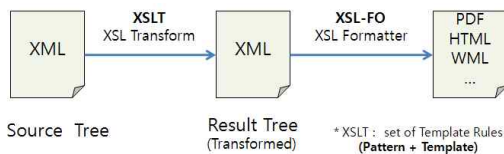
2.2 웹 정보의 동적관리를 위한 기법

2.2.1 XSL과 XPATH

웹 정보를 동적으로 표현할 수 있는 마크업 언어인 XML은 다양한 형태의 정보를 구조화하여 저장하고 유연하게 표현할 수 있는 효율적이고 안전한 데이터처리 방법이자 양식이다[6]. XML이 데이터와 구조정보를 제공하는 표현방법이라면, XSL(eXtensible Stylesheet Language)은 해당 문서가 어떻게 보여질 것인지에 대한 스타일시트를 제공한다.

XSLT(XSL Transformations)는 문서구조의 변환기능을 담당하며 XPATH는 XML PATH Language로서, 문서의 변환시 기존 문서에서 가져올 특정 노드를 가리키는 기능이 제공된다. XSLT는 같은 정보라 할지라도 여러 가지 형태에 따른 다양한 표현방법이 가능하도록 하는 구조 및 스타일 변환용 모듈로서 관리자의 직관적인 분석 및 보고용 자료작성에 쉽게 적용된다.

XSL-FO는 XML Formatting Objects 로서 문서형태를 변환하는 기능이 보장되었다. 따라서 이러한 XSLT를 이용하면 탐색된 전체 상황 정보 중 분석에 필요한 몇 가지 항목에 대해서 집중하고자 하는 경우나, 동일한 정보에 대하여 모니터링용 HTML, 문서작성용 PDF파일, 데이터 저장용 TXT등으로 변환하는 작업이 수월하다. <그림 2>는 XML문서에 위의 기법들을 이용하여 다양한 형태로 표현하고 변환하는 단계를 도식화하여 나타내고 있다.



<그림 2> XSLT를 이용한 XML의 다양한 표현 및 변환 방법

2.2.2 JMX(Java Management eXtensions)

JMX(Java Management eXtensions)는 JMAPI(Java Management API)에서 발전한 것으로 쉽게 네트워크 관

리에 필요한 구성요소를 개발할 수 있는 방법과 툴을 제시하고 있다[7]. JMX의 구조에서 핵심이 되는 컴포넌트는 관리객체로 확장을 지원하는 MBean, 객체에 대한 관리기능을 지원하는 MBean Server, 그리고 HTTP, RMI, SNMP 같은 관리 프로토콜을 이용하여 접속할 수 있도록 인터페이스를 제공하는 Protocol Adapter, 관리기능을 주체적으로 수행하는 JMX Manager가 있다[8-9]

2.2.3 AJAX

AJAX(Asynchronous Javascript and XML)자바스크립트와 XML을 이용하는 웹 브라우징 기술로서 특정한 기술 분야를 이르기보다 웹 2.0을 위한 동적인 웹 환경을 구현하기 위한 기술을 통칭하여 사용된다[10, 11]. 예를 들어 갱신되거나 수정된 정보를 현재의 웹 화면에 적용하기 위해서 브라우저를 재로딩하거나 페이지 이동을 하는 방식이 아니라, 클라이언트 측에서 필요한 정보를 반영하도록 처리한다. 보통 서버측에서는 SOAP 기반이나 XML이 서비스 프로토콜로 사용되며, 웹 서버의 응답을 처리하기 위하여 클라이언트 측에서는 자바스크립트를 이용한다[12].

2.3 실시간 시스템 vs. 준실시간 시스템

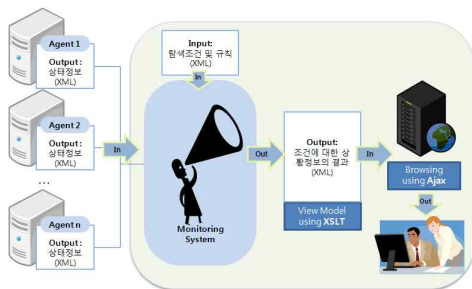
모니터링 시스템 환경은 관리자가 필요한 상태 정보를 신속히 탐색하여 제공할 수 있도록 해야 한다. 또한 정보시스템에 새롭게 추가된 어플리케이션을 모니터링하기 위해 모니터링 시스템은 재시작 없이 지속적인 탐색을 해야 한다. 분산 환경에서의 이중의 시스템에서 동작하는 개별 서비스 상태를 인식할 수 있어야 하며 요구 시간 안에 데이터를 탐색-가공-저장-표현이 가능하여야 한다. 그러나 복잡한 환경 속에서 대량의 정보를 탐색하여 표현을 하는 동안 시간지연이 발생할 수 있으므로 진정한 의미의 실시간 처리를 하기란 곤란하다. 따라서 최근에는 주기를 정하여 대량의 데이터를 탐색하여 임시로 저장해 둔 다음 가공한 후에 변환하여 표현하는 준실시

간(Semi-RealTime) 시스템에 대한 관심이 높다. 실시간 시스템과 준실시간 시스템이 갖는 장·단점이 존재하지만, 굳이 실시간으로 해결해야 할 이유는 없다. 준실시간 시스템은 시간 지연이 발생할 수 밖에 없는 환경적인 요소를 감안한 것에서 대두된 대안으로써, 대량의 데이터 분석 작업이 이루어지는 데이터웨어하우스 분야에서는 준실시간으로 처리되도록 하고 있다.

집 기능을 각각의 모듈로 구성하였다. 상태를 판단하고 관리자에게 모니터링 화면을 제공하는 모듈인 Manager Module은 JAVA Class로 구성되어 있다.

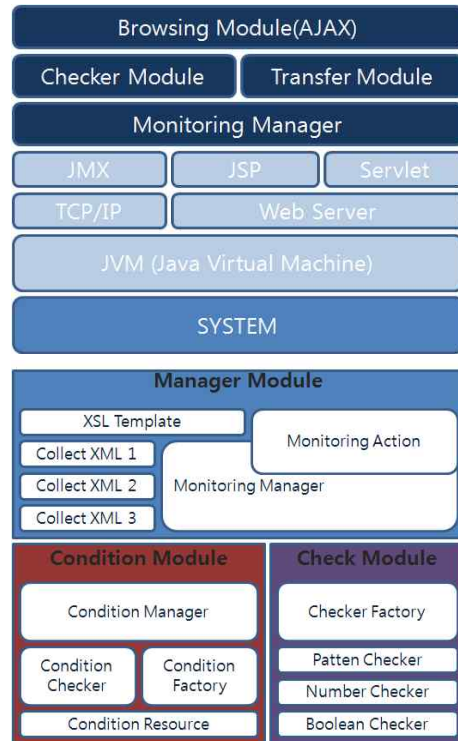
III. AJAX + XML 기반의 모니터링 시스템

본 논문에서 제안하는 모니터링 시스템의 개요는 <그림 3>과 같다. 각 서버로부터 서비스와 어플리케이션의 상황정보를 수집하여 XML로 저장한 후 모니터링 서버에게 전달한다. 이 상황정보는 운영시스템의 탐색규칙 및 조건에 의해 XSLT를 사용하여 재구조화하여 궁극적인 모니터링 모델인 뷰모델을 제공한다. 브라우징 모듈은 구현은 웹 환경에서 준실시간 관리와 분석이 유연하게 동작되도록 하고 모니터링으로 인한 시스템의 성능저하를 최소화하기 위해 AJAX를 이용하여 구현하였다.



<그림 3> AJAX + XML기반의 모니터링 시스템의 개요

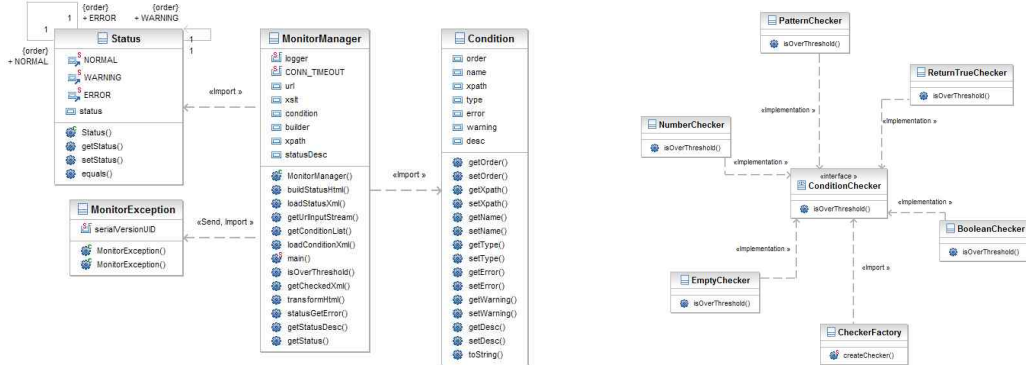
<그림 4>는 이에 필요한 기반시스템과 모듈의 구조도를 도식화하여 나타내며 <그림 5>는 본 논문에서 제안하는 시스템의 클래스 다이어그램을 나타낸다. 제안 시스템에서는 모니터링 대상과 모니터 서버와의 연결 복잡도를 최소화하기 위하여 상태 정보 및 모니터링 데이터 수



<그림 4> AJAX + XML기반의 모니터링 시스템의 구조도

주요 모듈의 기능과 역할은 다음과 같다.

- Manager Module : Manager Module은 각각의 서버로부터 데이터를 수집하여 XML로 저장한 후, Condition Module을 이용하여 <표 1>과 같이 주어지는 모니터링 조건에 기반하여 필요 정보를 구분해 낸다. 그 후, Checker Module을 사용하여 오류 및 이상여부를 검사하고 모든 정보들을 관리자 화면으로 전달한다. 여기서 수집되고 가공되는 정보들은 모두 XML로 구성되며 관리자용 화면은 관리용 XSLT 템플릿 데이터를 사용하여 생성된다. 최종 브라우징 화면은 AJAX을 기술을 이용하여 구성함으로써 동적인 응답속도를 최대화 하였다.



<그림 5> AJAX + XML기반 모니터링 시스템의 클래스 다이어그램

· Condition Module : Condition Module은 모니터링 목표와 대상의 특징이 기술되어 있는 템플릿 파일에 근거하여, Manager Module이 분석할 수 있는 형태로 변환시켜준다. 변환양식은 <표 1>에서 정한 태그를 이용하여 XML로 기록되도록 하며 주어진 조건들을 <그림 6>의 예와 같이 condition.xml의 형태로 XML 파일로 기록한다. 이 모듈에서는 이러한 화일에 기반하여 분석대상의 위치를 XPATH로 지정하고, 데이터 검증을 위해 검사기준 및 데이터항목을 구분하여 별도의 XML 파일을 작성한다. 이 때, 모니터링 환경에 새로운 서비스와 어플리케이션, 시스템 특징이 추가되면 시스템을 재시작하거나 변경내용을 매번 재프로래밍하는 것이 아니라 <그림 6>의 예와 같이 xml 파일

에 새로 확장된 내용을 추가해 주어 실시간으로 반영 되도록 구성되었다.

· Checker Module : Checker Module은 다양한 환경에서 서로 다른 형태로 수집되는 모니터링 데이터의 데이터 형식의 변환을 지원하고 검사하는 모듈이다. 모니터링 시스템 내부에서 사용하기 편한 형태로 변경하고 동일한 기준으로 검사할 수 있도록 변환하는 기능을 수행한다. 또한 이렇게 변환된 데이터의 접근성을 높이기 위해 Manager Module에서 동일한 인터페이스로 접근하여 현재 상태를 분석하게 하는 메서드를 제공한다.

제한 시스템에서는 DB의 대안으로 XML을 기본적인

<표 1> 모니터링 시스템의 분석 대상으로 설정되는 태그의 항목과 의미

태그	설 명
name	condition 항목의 이름
xpath	status.xml의 어느 항목의 값을 검사할 것인지의 xpath
type	설정된 검사 유형 · number : 숫자 값을 검사한다. 비정상 상태인 warning/error 태그의 값을 검사하여 warning/error 상태를 설정한다. · empty : xpath의 값이 존재하는지 검사한다. 이 때는 warning/error 태그가 불필요하며 존재하지 않을 경우 error 상태가 된다. · pattern : xpath의 값이 warning/error의 값과 일치하는지 확인한다.
warning	서비스의 상태를 warning로 설정할 경우의 값
error	서비스의 상태를 error로 설정할 경우의 값

정보 저장 매체로 사용하며, SQL 쿼리의 대안으로 XPATH 기능을 이용하여 XML로 저장되어 있는 내용 중 특정 데이터에 접근할 수 있도록 하였다.

```
<?xml version="1.0" encoding="UTF-8"?>
<conditions>
.....
<condition>
<name>힙 메모리 사용율</name>
<xpath>/routers/local/router/heapStatus/usage
</xpath>
<type>number</type>
<warning>80</warning>
<error>95</error>
</condition>
.....
</conditions>
```

<그림 6> condition.xml의 작성 예

<그림 6>의 예와 같이 XML로 정보를 표현하고 XPATH를 사용함으로써 정보의 접근성을 높이도록 구현하였고, 표현양식의 변화와 정보 저장형식의 변화에 능동적이고 유연하게 대처할 수 있도록 하였다. 또한 XPATH를 이용하여 정보의 저장위치를 지정하고 해당 위치에는 모니터링에 필요한 탐색정보를 데이터 표현 규칙에 따라 나타냄으로써 관리자의 직관적인 판단과 분석이 가능하도록 하였다. 본 논문에서 사용된 XML, XPATH, AJAX등의 기술들은 탐색된 상태 정보를 관리자의 다양한 분석목표에 따라 재구조화 할 수 있도록 하여 유연성을 높였다.

IV. AJAX + XML 기반의 모니터링 시스템 적용 및 실험

본 논문에서 제안한 시스템을 관리자의 실시간 모니터링 작업과 분석, 적절한 피드백이 요구되는 통신 서비스에서 적용하고 테스트 결과를 정리한다.

4.1 SMS 서비스 모니터링

SMS 서비스통신사는 급증하는 사용률에 비해 처리량과 관리력이 미비한 대표적인 서비스 중 하나이다. 일정 건수 이상으로 전송대기량이 많아지게 되면 서비스에 부하가 발생하게 되고, 오류감지 후 원인을 찾아 재시작부터 재전송에 이르는 지연시간은 사용자의 불만을 넘어서는 큰 사건이 될 것이다. 특히, 수신-송신자 측의 전송규약이나 상황에 변화가 생길 경우에는 즉각적인 상황 파악과 반응이 이루어져야 할 것이다.

4.1.1 서비스 환경

현재 운영 중인 여러 서비스들의 상태를 웹 인터페이스를 통해 XML 문서형태로 제공하며, 각 서비스들은 XSL을 이용하여 이 문서들을 HTML로 변환하여 현재 운영되는 서비스의 상태를 웹 브라우저를 통해 확인한다. <표 2>는 SMS 서비스의 상태를 정리하여 나타내고 있다. 이 때, 일반적으로 각 서비스 별 접속해야 할 URL이 다르므로 전체 서비스의 상태를 조회하기 위해서는 모든 서비스의 상태 페이지들을 일일이 각각 접속해야 하는 불편함이 있다. 본 시스템은 이러한 URL들을 한 화면에서 통합 모니터링이 가능하도록 구현되었다.

- 제안 시스템에서 각 디렉토리의 역할은 다음과 같다.
- bin : 시작 및 정지 스크립트
 - logs : 로그

서비스명		상태 XML URL
SMS Agent 3.0	SMS CPN2	http://192.168.89.102:8083/status
	SMS NOTI	http://192.168.89.102:8084/status
	SMS PAYM	http://192.168.89.102:8085/status
	SMS REAL	http://192.168.89.102:8086/status
	SMS SFA	http://192.168.89.102:8087/status
SMS Agent 3.1	SMS NOTI2	http:// 192.168.89.102:8088/status
MMS	MMS	http://192.168.89.102:8081/status
	MMS SSL	http://192.168.89.102:8011/status

<표 3> SMS 서비스 모니터링 시스템의 분석 대상

검사 대상	검사 데이터의 XPATH	데이터 타입	경고기준 (Warnng)	위험기준 (Error)
힙 메모리 사용율	/routers/local/router/heapStatus/usage	number	80 이상	95이상
전송대기 큐의 크기	/status/queueStatus/incomingQueue	number	4000 이상	5000이상
전송 큐의 크기	/status/queueStatus/sendQueue	number	2000 이상	3000이상
수신서버의 상태	/status/currentConnection/Recieve	String	DISABLE 상태	DISABLE 상태
저장모듈의 상태	/routers/local/router/modules/module [name='DatabaseLogger']/name	boolean	false	false

- webapp : 각 데몬의 XML, HTML
- tomcat : Tomcat 설치위치
- \$. /bin/start(stop) : 시작(정지)스크립트 실행

4.1.2 상태 검사 조건의 설정

현재 운영되는 서비스의 상태 중 어떤 항목의 값을 검사하여 서비스의 상태를 결정할 것인가를 설정한다. 본 실험에서는 위의 <표 2>의 운영상황에 따라 힙 메모리, 전송 큐, 수신서버 등의 대상을 모니터링 한다. <표 3>은 검사 대상별 XPath 정보와 경고 및 위험기준을 나타낸

다. 상태 정보가 저장된 XML 문서에서 검사 항목의 값을 체크하여 현재 서비스의 상태를 정상(Normal)/경고(Warning)/위험(Error)의 세가지 상태로 정하고 관리자가 현재 서비스의 상태를 한 눈에 확인할 수 있게 하였다.

4.2 적용 화면

아래의 <그림 7>은 SMS 서비스들에 대해 앞서 정한 환경으로 실험한 결과를 보인다. 적용 결과는 웹 문서로

■ SMS Agent3.0

서비스명	전송대기불 대기 메시지	큐 대기 메시지	JVM 메모리	상태	설명
SMS CPN2	0	0	2.36%	normal	
SMS NOTI	0	0	2.34%	normal	
SMS PAYM	0	0	2.56%	normal	
SMS REAL	0	0	2.45%	normal	
SMS SFA	0	0	2.44%	normal	

(a) 정상 동작

■ SMS Agent3.0

서비스명	전송대기불 대기 메시지	큐 대기 메시지	JVM 메모리	상태	설명
SMS CPN2	0	0	85%	warning	힙메모리 사용률(85) >= 85
SMS NOTI	0	0	85%	warning	힙메모리 사용률(85) >= 85
SMS PAYM	0	0	85%	warning	힙메모리 사용률(85) >= 85
SMS REAL	0	0	85%	warning	힙메모리 사용률(85) >= 85
SMS SFA	0	0	85%	warning	힙메모리 사용률(85) >= 85

(b) Warning 상태 : JVM 메모리 사용률이 85%일 경우

■ SMS Agent3.0

서비스명	전송대기불 대기 메시지	큐 대기 메시지	JVM 메모리	상태	설명
SMS CPN2	0	0	95%	error	힙메모리 사용률(95) >= 85
SMS NOTI	0	0	95%	error	힙메모리 사용률(95) >= 85
SMS PAYM	0	0	95%	error	힙메모리 사용률(95) >= 85
SMS REAL	0	0	95%	error	힙메모리 사용률(95) >= 85
SMS SFA	0	0	95%	error	힙메모리 사용률(95) >= 85

(c) Error 상태 : JVM 메모리 사용률이 95%일 경우

<그림 7> JVM 메모리 사용률의 상황정보 인식을 위한 SMS 모니터링 시스템

나타나게 되며 관리자의 직관적인 상황파악이 용이하도록 구성되었다. 이기종의 SMS서버에서 생성되는 정보 중 메모리 상태에 대한 변화를 분석하기 위한 탐색정보와 규칙을 정하여 상황정보를 수집한 후의 모니터링 화면이다. 아래 그림의 두 번째 필드에서 '전송테이블 대기 메시지'는 전송테이블에서 에이전트가 읽어(Fetch)오지 못한 메시지의 건수를 의미한다. '큐 대기 메시지'는 에이전트가 전송테이블에서 읽어와 전송대기 큐에 존재하는 메시지의 건수를 의미한다. 'JVM 메모리'는 에이전트가 사용하는 JAVA HEAP 메모리 영역의 사용률을 나타낸다.

<그림 7>에서는 메모리 영역의 상태의 정확한 상황 조회를 위하여 초기 시스템의 전송건의 요청이 없는 상태에서 JVM 메모리를 임의로 변화시켜 운영상황의 변화

를 살펴도록 하였다.

<그림 8>은 대기 메시지의 상황정보의 조회를 위하여 최소한의 내용으로 전송 건수를 늘려 테스트 한 결과를 보인다. Webapp/conf/_condition.xml 파일에는 공통적으로 JVM 메모리가 80% 이상인 경우 경고 상태, 95% 이상인 경우 위험 상태, 그리고 대기메시지의 건수에 대한 조건을 설정하였다. 또한 필수적으로 동작 중이어야 하는 모듈들이 수행중인지 검사하여 동작 중이지 않은 경우 '위험 상태'로 표시하도록 설정하였다. 해당 XML 파일에서 '경고/위험'을 결정하는 기준 값을 설정하도록 하였으며, 업무 성격 및 전송량 등을 파악하여 결정하여야 한다. 일반적인 오류 상황 조건 및 발생 시 확인 방법을 정리하면 다음과 같다.

■ SMS Agent3.1				
서비스명	대기 메시지	JVM 메모리	상태	설명
SMS NOTI2	0	6.00%	normal	

■ MMS Agent				
서비스명	대기 메시지	JVM 메모리	상태	설명
MMS	0	6.00%	normal	

■ MMS SSL				
서비스명	대기 메시지	JVM 메모리	상태	설명
MMS SSL	0	6.00%	normal	

(a) 정상 동작

■ SMS Agent3.1				
서비스명	대기 메시지	JVM 메모리	상태	설명
SMS NOTI2	4000	6.00%	warning	전송대기큐 크기(4000) >= 4000

■ MMS Agent				
서비스명	대기 메시지	JVM 메모리	상태	설명
MMS	4000	6.00%	warning	전송대기큐 크기(4000) >= 4000

■ MMS SSL				
서비스명	대기 메시지	JVM 메모리	상태	설명
MMS SSL	4000	6.00%	warning	전송대기큐 크기(4000) >= 4000

(b) Warning 상태 : 전송 대기 메시지가 4000건 이상인 경우

■ SMS Agent3.1				
서비스명	대기 메시지	JVM 메모리	상태	설명
SMS NOTI2	5000	6.00%	error	전송대기큐 크기(5000) >= 4000

■ MMS Agent				
서비스명	대기 메시지	JVM 메모리	상태	설명
MMS	5000	6.00%	error	전송대기큐 크기(5000) >= 4000

■ MMS SSL				
서비스명	대기 메시지	JVM 메모리	상태	설명
MMS SSL	5000	6.00%	error	전송대기큐 크기(5000) >= 4000

(c) Error 상태 : 전송 대기 메시지가 5000건 이상인 경우

<그림 8> 대기메시지 상태의 상황정보 인식을 위한 SMS 모니터링 시스템

- 95% > 메모리 사용률 >= 80% : 자바의 힙 메모리 사용률이 80% 이상 되면 warning 상태로 인식하여 알람을 지정한다.
- 사용률 >= 95% : 이후 지속적으로 메모리가 증가하여 95% 초과시 error 상태로 인식하여 알람을 지정한다.
- 5000건 > 전송대기큐 >= 4000건 이상 : 경고알람
- 전송대기큐 >= 5000건 이상 : 에러알람

특히, JVM 메모리가 95%를 초과하여 위험알람이 발생된 이후에는 OutOfMemoryError가 발생하여 서비스가 중지될 위험이 있으므로 정상화를 위한 지속적인 모니터링이 필요하다.

V. 결론

실시간으로 동작해야 하는 서비스와 정보시스템들은 최소한의 관리자의 개입으로도 보다 안정된 상태에서 높은 신뢰도와 정확성을 만족하며 운영되어야 한다. 본 논문에서는 AJAX와 XML기술을 이용한 모니터링 시스템을 설계 및 구현하여 지속적인 모니터링과 처리량에 따른 오류가 빈번히 일어나는 SMS 시스템에서 적용하였다. 가장 모니터링 부하가 높은 메모리와 대기건수에 대하여 이중의 서버에 쌓인 전송상태에 대한 정보를 실시간으로 읽어와 직관적인 정보과약을 위한 동적인 페이지로 나타나게 된다. 관리자는 빠른 상황정보의 인식이 가능하며, 매뉴얼의 규칙에 의해 각 전문분야의 담당자에게 전달된 보고(reporting)문서가 생성된다. 이와 같은 SMS 서비스 뿐만 아니라 유비쿼터스 환경의 컴퓨팅 서비스와 상황인지 서비스등과 같은 최근의 시스템을 구현하기 위해서 많은 연구들이 진행되고 있다. 단지 동작만 되는 시스템이 아니라 서비스의 질적인 성장을 이끌려면 시스템 내부에서 수행되고 있는 처리 과정의 오류를 살피고 결과를 분석해 내어 이전과정으로 피드백 할 수 있는 더 향상된 시스템을 지향하도록 발전해가야 할 것이

다. 추후 다양하고 이질적인 시스템에서 생성되는 중요한 모니터링 정보들에 대해 효율적인 표현방법과 직관적인 분석이 가능하도록 발전시키는 문제에 대해 연구되어야 할 것이다.

참고문헌

- [1] C. Steigner, K. Wilke, "Multi-Source Performance Analysis of Distributed Software," the Communication networks and Distributed Systems Modeling and Simulation Conference. January 2002.
- [2] 권성현, 이병훈, 김재훈, 조위덕, "유비쿼터스 시스템을 위한 실시간 모니터링 에이전트," 정보과학회 논문지, 제14권 제 8호. 2008.
- [3] Y. N. Lee, J. T. Lee and M. K. Kim, "Multi-Agent based Community Computing System Development with the Model Driven Architecture," Autonomous Agents and Multi-Agent Systems, May 2006.
- [4] 권성주, 최재영, 이지수, "계층형 구조를 기반으로 한 모니터링 시스템," 정보과학회논문지:시스템 및 이론, 제 33권 제 7호, 2006, pp. 440~447.
- [5] 박준성, 이귀로, 조진성, "uPetCare : 웹2.0을 이용한 유비쿼터스 펫 케어 시스템", 한국정보과학회 논문지, 제 15권 제 4호, 2009, pp. 0260~0264.
- [6] W3C, "Extensible Markup Language," <http://www.w3c.org/XML>.
- [7] K. Arnold and J. Gosling, The Java Programming Language, Addison-Wesley, 1996.
- [8] SUN, "Java Management(TM) Extension (JMX) Home Page," <http://java.sun.com/products/JavaManagement/>.
- [9] J. P. Martin-Flatin, L. Bovet, and J. P. Hubaux.

“JAMAP: a Web-Based Management Platform for IP Networks,” DSOM’99, October 1999.

- [10] AJAX, <http://www.ajax.org>
- [11] Asynchronous JavaScript Technology and XML (Ajax) With the Java Platform, <http://java.sun.com/developer/technicalArticles/J2EE/AJAX/>
- [12] SUN, “Java(TM) Security API,” <http://java.sun.com/security/>

■ 저자소개 ■



최 윤 정
Choi, Yun Jeong

2009년 3월~현재
서일대학교 정보통신과 강의전담
2007년~2008년
서강대학교 컴퓨터학과 Post. Doc
2007년 2월 이화여자대학교
컴퓨터공학과(공학박사)
2001년 8월 이화여자대학교
컴퓨터공학과(공학석사)
1997년 2월 서원대학교 전자계산학과(이학사)

관심분야 : 인공지능, 기계학습, 온톨로지,
상황정보인식, 유비쿼터스
센서네트워크
E-mail : cris@seoil.ac.kr



박 승 수
Park, Seung Soo

1991년~현재
이화여자대학교 컴퓨터공학과 부교수
1991년 미국 켈사스대학 컴퓨터학과 조교수
1988년 미국 텍사스 오스틴 대학(박사)
1976년 한국과학기술원 수학과(석사)
1974년 서울대학교 수학과(공학사)

관심분야 : 인공지능, 데이터마이닝, 상황인식,
유비쿼터스, 바이오인포매틱스
E-mail : sspark@ewha.ac.kr

논문접수일 : 2009년 10월 29일
수 정 일 : 2009년 11월 15일
게재확정일 : 2009년 11월 22일