

## JMX를 이용한 통합 망 관리 시스템 설계

김대영<sup>o</sup> 이종언 차시호 조국현  
광운대학교 컴퓨터학과  
{dykim<sup>o</sup>, jelee, sihoc khcho}@cs.kw.ac.kr

### Design of integrated network management system using JMX

Daeyoung Kim<sup>o</sup> Jongeon Lee Siho Cha Kukhyun Cho  
Dept. of Computer Science, Kwangwoon University

#### 요 약

기존 원격 통신 네트워크나 서비스를 위한 관리는 통합된 관리 모델(SNMP/SMI)을 사용함으로써 다양한 기반 구조의 통합을 가능하게 하였다. 그러나, 음성과 데이터의 통합, 유선과 무선의 융합, 사업자간 서비스 영역 구분으로 인하여 기존 관리 시스템은 비능률적이고, 복잡하며, 구현과 큰 네트워크에 적용하기 어렵다. 이를 개선하기 위하여 SUN에서 네트워크 관리 기술인 JMX(Java Management Extensions)를 제공하고 있다. JMX는 어플리케이션과 네트워크 관리 및 모니터링을 위한 아키텍처이며, 현존하는 관리 및 모니터링 시스템을 통합하는 수단을 제공하여 준다. 따라서, 본 논문에서는 JMX를 활용하여 통합 네트워크 관리를 할 수 있는 시스템을 설계한다.

#### 1. 서 론

차세대 네트워크는 기존의 공중사설망, 이동전화망, 패킷망을 통합하여 수용할 수 있는 패킷 기반의 개방형 통신망이며, 네트워크 구조를 서비스 계층, 제어 계층, 전달 계층으로 분리하여 정의하고 있으며, 각 계층간의 인터페이스는 개방 표준인터페이스를 정의하여 제공하고 있다. 이런 인터페이스의 정의는 Parlay 그룹에서 정의하고 있으며, 작업그룹에 의해 생성된 UML(Unified Modeling Language)은 JAIN(Java API for Integrated Network)에 의해 API(Application Program Interface)로 제공한다. 이러한 API로 인하여 다양한 네트워크(PSTN, Broadband, Wireless, IP, Satellite)에서 응용을 동작할 수 있도록 하고, 벤더와 서비스 제공자들에게 독립적으로 동작할 수 있도록 한다[9].

하지만, 네트워크 통합을 추구하면서, 문제가 되고 있는 것이 네트워크 관리이다. 기존 원격 통신 네트워크나 서비스를 위한 관리는 통합된 관리 모델(SNMP/SMI)을 사용함으로써 다양한 기반 구조의 통합을 가능하게 하였다. 그러나, 음성과 데이터의 통합, 유선과 무선의 융합, 사업자간 서비스 영역 구분으로 인하여 현 관리 시스템들은 비능률적이고, 복잡하며, 구현 및 큰 네트워크에는 적용하기 어렵다[10]. 이를 실현하기 위하여 SUN에서 네트워크 관리 기술인 JMX(Java Management Extensions)를 제공하고 있다. JMX는 자바 프로그래밍 언어에서의 어플리케이션과 네트워크 관리 및 모니터링을 위한 아키텍처, 디자인 패턴, API, 그리고 서비스층을 정의한다. 또한, JMX 스펙은 자바 코드에 적용해 스마트 자바 에이전트를 만들고, 분산 관리 미들웨어 및 관리자를 구현하며, 이들 솔루션을 현존하는 관리 및 모니터링

시스템으로 부드럽게 통합하는 수단을 업계의 모든 자바 개발자들에게 제공한다[5][10]. 또한, 현존하는 표준 관리 및 모니터링 기술에 대한 여러 Java API가 JMX 스펙을 언급하고 있다. 따라서, 본 논문에서는 JMX를 활용하여 통합 네트워크 관리를 할 수 있는 시스템을 설계한다.

#### 2. 관련연구

##### 2.1 JMX(Java Management Extensions)

JMX는 관리와 관련한 단일하면서도 개방된 기술로 모든 산업에 곧바로 배포해 적용할 수 있도록 정의한 규격으로 기본적으로 레거시 시스템에 적용하기 쉽고, 새로운 관리 솔루션을 구현한 뒤에 이를 추가하는 좋은 확장성을 가지고 있다. 또한 JMX는 DMTF(Distributed Management Task Force)라는 표준화 단체에서 정의하는 WBEM(Web-Based Enterprise Management) 표준을 지키고 있으며, 웹 기반 분산, 모듈화 관리 장치나 어플리케이션 네트워크 관리 등에 대한 소프트웨어를 쉽게 구축할 수 있는 도구를 제공한다. JMX는 기본적으로 그림 1과 같이 세 개의 계층으로 구성되어 있다[5].

##### - 적용 레벨 (Instrumentation level)

JMX로 관리 가능한 자원을 구현하기 위한 스펙을 제공한다. 이러한 자원은 어플리케이션이나 서비스의 구현, 장치, 사용자 등이 될 수 있다. 이러한 자원은 JMX 호환 어플리케이션이 관리할 수 있도록 적용되어야 한다. 이렇게 주어진 자원의 적용은 하나 이상의 표준이거나 동적인 MBean(Managed Beans)에 의해 제공된다.

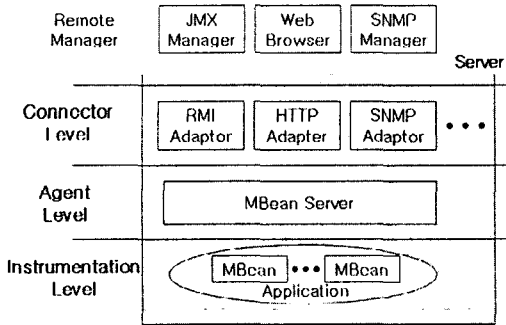


그림 1. JMX의 구조

- 에이전트 레벨 (Agent level)

에이전트를 구현하기 위한 스펙을 제공한다. JMX로 관리 가능한 자원을 관리하는 표준화된 에이전트를 정의하기 위해 적용 레벨 위에 개발되어 적용 레벨을 사용한다. 또한, MBean 서버와 MBean 들을 다루는 여러 서비스들(Timer, Monitoring, 등등)로 구성된다. 따라서, JMX 에이전트는 JMX로 관리 가능한 자원이 상주한 머신에 자바 가상 머신이 사용 가능하다면, 같은 머신에 임베디드할 수 있고, 사용할 수 없는 환경이면 중개자 역할을 할 수 있다.

- 분산 서비스 레벨 (Distributed services level)

JMX 관리자를 구현하는 인터페이스를 제공한다. 즉, 에이전트나 에이전트의 계층구조와 함께 작업할 수 있는 관리 인터페이스와 컴포넌트를 정의한다. 이들 컴포넌트들은 커넥터를 통해 에이전트와 JMX로 관리 가능한 자원을 투명하게 상호작용하는 관리 어플리케이션을 위한 인터페이스를 제공하며, 고수준 관리 플랫폼에서부터 수많은 JMX 에이전트까지 관리 정보를 분산시킬 수 있다.

그림 1과 같이 관리자 레벨과 다른 에이전트 및 적용 레벨의 조합은 완전한 관리 솔루션을 디자인하고, 개발하는데 필요한 완전한 아키텍처를 제공한다. 따라서, JMX 기술은 유일하게 인식성, 관리 기능의 on-demand 배치, 동적 및 이동성 서비스, 그리고 보안을 모두 제공해 준다.

2.2 JDMK (Java Dynamic Management Kit)

JDMK는 관리 어플리케이션을 설계하고 구현하기 위한 개발 툴이며 자바 API(Application Programming Interface)이다. 또한, 자바 기술기반 어플리케이션을 통하여 자바 관리 오브젝트의 관리를 위한 프레임워크를 제공하며, 분산 관리 시스템을 설계하기 위한 완벽한 아키텍처를 제공한다. [2]

JDMK는 SNMP 에이전트 개발과 관리자를 위한 툴킷을 제공한다. 이 툴킷은 SNMP MIB 컴파일러, 자바 소스 코드로 SNMP MIB를 컴파일하기 위해 사용하는 mibgen 툴과 기존 connector와 함께 사용하기 위한 프락시 오브젝트 생성자인 proxygen 툴이 포함되어 기존

프로토콜과 함께 사용할 수 있다[1].

3. 통합 네트워크 관리 시스템 설계

3.1 설계 원칙

통합 네트워크 관리 시스템을 설계하기 위하여 고려할 점은 다음과 같다.

- 네트워크 통합 관리를 위하여, 어떤 네트워크에서든 지 관리를 할 수 있어야 한다. 따라서, 하위 계층을 분리 시켜야 하며, 하위 계층과 통신을 하기 위한 인터페이스가 필요하다.
- 기존 SNMP와 같은 프로토콜과 통합이 가능한 구조가 되어야 한다.
- 여러 서비스들이 지원 가능한 구조를 가져야 한다.
- 어플리케이션이든지 하드웨어이든지 모두 관리를 할 수 있는 구조를 가져야 한다.
- 응용 계층에서 자원 구성들에 대한 정보를 가지고 있어야 한다.

위와 같은 설계 원칙을 따르도록 시스템이 설계되어야 한다. 본 논문에서 설계한 시스템은 이 모두를 만족시켜주는 시스템이다.

3.2 통합 네트워크 관리 시스템 설계

그림 3은 본 논문에서 설계한 구조이며, 3.1절에서 제시한 설계 원칙을 따르는 통합 네트워크 관리 시스템이다.

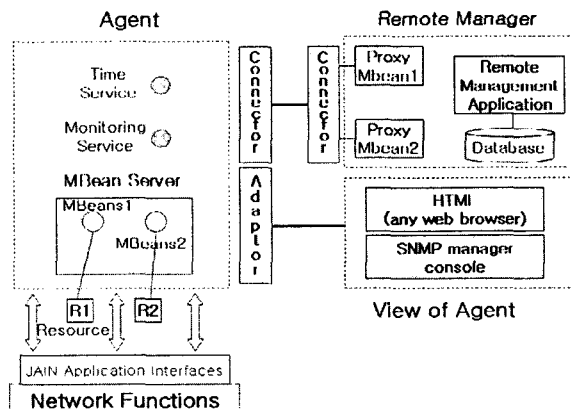


그림 3. 통합 네트워크 관리 시스템

여러 네트워크를 관리하기 위해서 하위 계층과는 JAIN 어플리케이션 인터페이스를 통하여 통신하여 어플리케이션 계층에서 자원을 구성할 수 있도록 하였다. JAIN은 표준화된 자바 인터페이스를 통해 개방 소프트웨어

위치 구조를 구축하고, 서비스를 제공하는 기술로, 네트워크 기반의 로직과 서비스 기반 로직을 분리해 어떤 망이든 서비스 제공이 가능하다.

그림 3에서 에이전트는 MBean 서버에 등록되어 있는 MBean들을 관리하기 위해 time, monitoring, alert, event등의 서비스들을 지원해 준다. 또한, 다른 프로토콜(SNMP, HTTP)과 함께 통신하기 위해서는 JMX adaptor를 사용하며, 다른 전송기술(Java RMI)등과 연결하기 위해서는 connector를 사용함으로써 쉽게 확장이 가능하고, 존재하는 기존 기술들과 독립적으로 통신이 되도록 하였다. 또한, 프락시는 JDK에서 제공되는 proxygen들을 사용하는데, proxygen은 기존 connector와 함께 사용하기 위한 프락시 오브젝트로서, 원격에서 MBean을 관리할 수 있도록 해주며, 모든 통신을 핸들링하게 해준다.

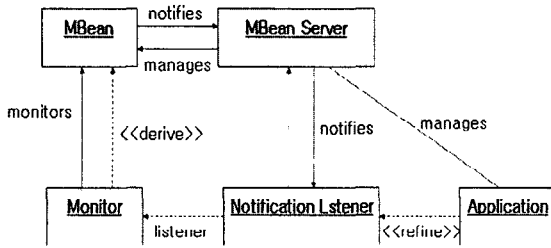


그림 4. 시퀀스 다이어그램

그림 4는 MBean 서버를 통하여 어플리케이션에 의해 관리되는 MBean을 보여주는 시퀀스 다이어그램으로 MBean을 지속적으로 모니터링 하다가, MBean의 상태가 바뀌면, 등록된 리스너에게 통보해주는 그림이며, 그림 5는 좀 더 세분화 하여 관리 동작이 원격 관리 어플리케이션으로부터 에이전트 측의 MBean으로 전달되는 것을 보여주고 있다. 관리 어플리케이션이 직접적으로 프락시 오브젝트의 getState 메소드를 invoke하고, 프락시 오브젝트는 Connector 인터페이스를 통해 에이전트에게 request를 전송한다. 그러면, 에이전트의 Connector 인터페이스가 이 요청을 받아서 MBean 서버를 통하여 MBean의 상태를 읽는 과정을 보여준다. 그 다음에 역으로 가는 것도 동일한 방법을 통하여 전달된다. 또한 MBeans1에서 이벤트가 발생하면, JMX Agent를 통해 원격 관리자에게 이벤트를 알리게 된다.

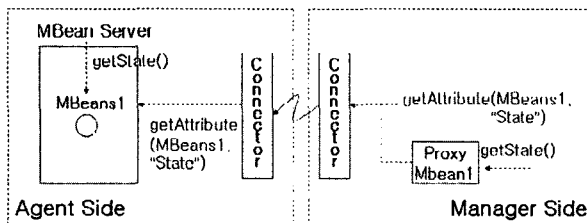


그림 5. 원격 관리 동작

#### 4. 결론 및 향후 과제

기존의 통신망은 유·무선 망이 분리된 수직적인 망 구조로 이질적인 망들이 혼재하여 존재하고 있으며, 이러한 이유로 타 망과의 연동이 취약하여 망의 개방화가 용이하지 않았지만, 본 논문에서 설계한 시스템은 네트워크 계층과 응용 계층을 서로 분리시킴으로써 각 기술이 독립적으로 발전할 수 있도록 하여 기술의 발전으로 인해 변화하는 네트워크와는 무관하게 서비스는 독립적으로 개발되고 이용할 수 있도록 하였다. 즉, 다양한 네트워크에서 어플리케이션을 동작할 수 있게 하고, 벤더와 서비스제공자들에게 독립적으로 동작할 수 있도록 하여 통합 네트워크를 관리할 수 있는 구조를 갖게 하였다.

향후 과제로는 미들웨어를 사용하여 네트워크 관리뿐만 아니라 OSS(Operation Support System)를 지원(QoS, 과금)하는 연구를 할 것이다.

#### [참고문헌]

- [1] Sun Microsystems Inc. "Java Dynamic Management Kit 5.1 Tools Reference Guide" 2004.
- [2] Sun Microsystems Inc. "Java Dynamic Management Kit 5.1 Getting Started Guide" June 2004.
- [3] Julio Guijarro, HP Labs. "Framework for managing large scale component based distributed application using JMX" May 2002.
- [4] Bahman Kalali, "Management of EJB Components Using JMX" March 15, 2002.
- [5] Sun Microsystems Inc. "Java Management Extensions White Paper" revision 01, June 1999.
- [6] Tony G. Thomas "JMX Boosts J2EE Application Management" 2002.
- [7] Victor A. Villagra, "An Approach to the Transparent Management Instrumentation of Distributed Applications" 2001.
- [8] Vijay Machiraju, Akhil Sahai, And van Moorsel, "Web Services Management Network" 2002.
- [9] R.Bhat, R.Gupta, "JAIN Protocols APIs", IEEE Communications Magazine, January 2000.
- [10] Benjamin G. Sullins, Mark B. Whipple "JMX In Action" 2001.
- [11] H.Kreger "Java Management Extensions for application management" IBM Systems Journal, Vol 40, No 1, 2001
- [12] Frank Moreno, "Total e-Business Management : managing Application Server Components" EAI Journal, March 2002
- [13] <http://mx4j.sourceforge.net/> MX4j's Homepage, open source JMX for Enterprise Computing