

무선 환경에서 분산 자원 관리를 위한 에이전트의 구현

허지훈*, 류승기, 윤여환, 강원의

*한국건설기술연구원

e-mail:temujin@kict.re.kr

Implementation of Agent for Management of Distributed Resources in Wireless Environment

Ji-Hoon Heo*, Seung-Gi Ryu, Yeo-Hwan Yoon, Won-Eui Kang

*Korea Institute of Construction Technology

요 약

인터넷 비즈니스의 급성장에 따라 인터넷의 이용 및 의존도가 높아지고 인터넷 사용자나 인터넷 서비스 제공자(ISP) 모두 안정된 서비스를 요구하고 있다. 이와 함께 전산 시스템 관리와 장애 예방 등의 데이터 보호도 중요한 요소가 되었고 언제 어디서나 시스템 성능을 모니터링하고 운용되는 애플리케이션을 제어하고자 하는 사용자들이 증가하였다. 본 논문에서는 모바일 환경하에서 JMX(Java Management Extensions)를 이용하여 분산된 자원과 서비스를 관리하는 에이전트를 구현한다. 본 논문에서 제안하는 시스템을 통해 시스템 관리자는 자신의 모바일 단말기를 통해 시간, 장소에 구애 받지 않고 시스템을 안전하게 관리할 수 있다.

1. 서론

오늘날 인터넷과 웹이 세계화되어감과 동시에 점점 더 복잡해지고 인터넷 활용 인구가 늘어남에 따라 시간적인 제약없이 연속된 컴퓨터 서비스가 필요하게 되었다[1]. 이와 같은 인터넷 비즈니스의 발달로 컴퓨터와 네트워크는 더욱 복잡하고 다양해짐에 따라 중앙 집중적 작업 환경을 분산 작업 환경으로 바꾸게 하였고, 분산 환경에서 작업하면서 관리자가 관리해야할 전산 시스템도 크게 증가하게 되었다. 컴퓨터 시스템은 매우 복잡하고, 변화하므로 항상 CPU, 메모리와 같은 성능이나 프로세스 상태를 주기적으로 모니터링 해야 한다.

최근 무선 이동통신기기의 발달로 모바일 단말기를 이용한 원격 제어 시스템이 발달하고 있어, 시간과 공간의 제약이 없는 모바일 단말기를 이용한 원격지 시스템 관리의 필요성이 증가하고 있다[2,3]

본 논문에서는 JMX 기술을 사용하여 분산 자원 관리를 위한 원격 제어 시스템을 제안하고 원격지 시

스템에 직접 접속을 해야 할 경우를 위해 원격지 터미널 접속에 대한 방법을 제안한다. 그리고 클라이언트 환경은 모바일 단말기를 사용하여 시간적, 공간적 제약을 없앴기에 시스템 관리자는 자신의 모바일 단말기를 이용해 분산된 원격지 시스템을 모니터링 및 제어 하고 문제가 발생 되었을 때 즉각적인 처리가 가능하다.

2. 모바일 원격 제어 시스템의 구조

JMX는 썬 마이크로시스템즈에서 제안한 자원 관리 프레임워크이다[5]. JMX는 동적인 관리라는 시장의 요구와 관리 솔루션의 설계자와 개발자들에게 적절한 툴을 제공하기 위해서 개발되었다. 그리고 관리 대상과 관리하는 시스템간의 결합이 유연해진다. JMX는 Instrumentation level, Agent level, Distributed services level로 구성되어 있다.

- Instrumentation level : JMX는 이 레벨에서

MBean(Managed Bean)을 통해 리소스를 관리할 수 있게 한다. 여기서 리소스는 애플리케이션, 서비스의 구현과 디바이스 등이 될 수 있으며 이 레벨은 클라이언트에 해당된다.

- Agent level : 관리 대상 시스템에서 실행되며 에이전트는 직접 리소스를 관리하고 원격의 관리 시스템이 그 리소스들을 다룰 수 있도록 한다.
- Distributed services level : 관리 대상 시스템에서 실행되며 JMX매니저를 구현하기 위한 인터페이스를 제공한다. 높은 수준의 프로토콜을 제공하거나 여러 에이전트의 관리 정보를 통합해 사용자의 논리적인 뷰를 제공한다.

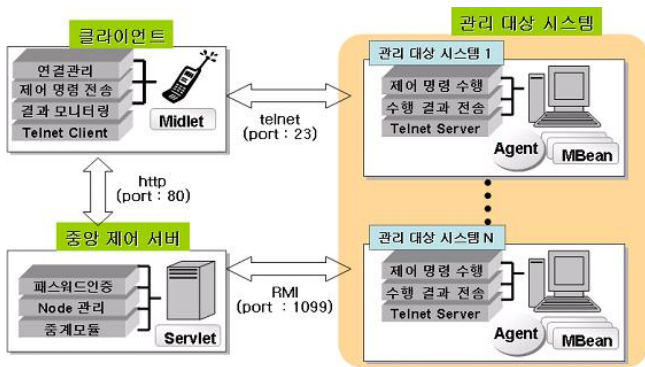


그림 1. 모바일 원격 제어 시스템의 구조

본 논문에서 제안하는 시스템 구조는 그림 1과 같이 표현되며 클라이언트, 중앙 제어 서버, 관리 대상 시스템으로 구성된다. 분산 시스템의 가장 두드러진 특징은 다수의 노드들이 존재하며 이들을 중앙에서 관리할 필요가 있다는 점이다[5]

클라이언트인 모바일 운영자는 자신의 모바일 단말기로 http로 중앙 제어 서버에 인증을 요청 한다. 인증이 되면 클라이언트는 중앙 제어 서버에 시스템 제어 요청을 하게 된다. 관리 대상 시스템에 직접 접속을 해야 할 경우 telnet 프로토콜을 통해 원격지 터미널에 접속을 해서 관리할 수 있다.

중앙 제어 서버는 모바일 단말기에서 요청받은 명령을 관리 대상 시스템의 Agent에 요청을 전달하고, 관리 대상 시스템으로부터 받은 결과를 모바일 단말기에 전송한다. 중앙 제어 서버와 Agent와의 통신은 자바 RMI(Remote Method Invocation)의 기본 통신 포트인 1099를 사용한다. 자바 RMI는 분산 시스템 환경하에서 서로 다른 하드웨어와 소프트웨어 플랫폼에서 발생하는 문제를 투명하게 처리하고 자바 가상 머신 위에서 작동중인 클라이언트 응용프로그램

이 원격 자바 가상 머신에서 구현된 서버 객체를 로컬 객체와 같은 방법으로 호출해서 사용할 수 있도록 지원해 준다[6].

관리 대상 시스템은 중앙 제어 서버로부터 요청받은 명령을 Agent가 MBean에 명령을 전달하고 MBean은 명령을 수행하게 된다. 그리고 수행 결과를 Agent에 결과를 전송하고, Agent는 그 결과를 다시 중앙 제어 서버로 보낸다. 클라이언트가 터미널 접속을 시도할 경우 텔넷 서버가 사용자 인증을 거친후 접속을 허용한다.

2.1 클라이언트

클라이언트는 자신의 모바일 단말기를 통해 중앙 제어 서버로 연결을 할 것인지 관리 대상 시스템으로 터미널 접속을 할 것인지를 선택한다. 클라이언트 구성은 그림 2와 같다.

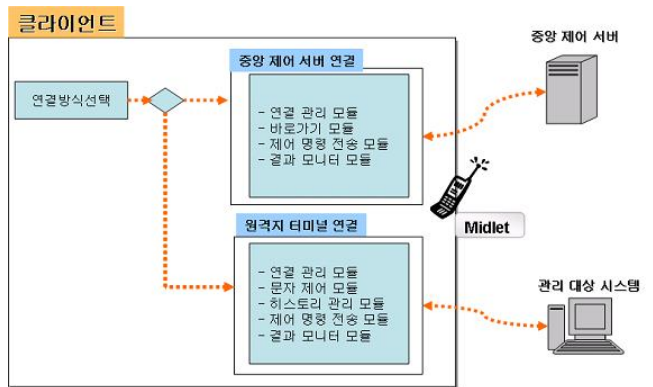


그림 2. 클라이언트 구성

중앙 제어 서버 연결은 연결 관리 모듈, 바로가기 모듈, 제어 명령 전송 모듈, 결과 모니터 모듈로 구성되며 클라이언트가 중앙 제어 서버로 연결해 여러 관리 대상 시스템을 모니터링 및 제어할 수 있게 한다.

원격지 터미널 연결은 연결 관리 모듈, 문자 제어 모듈, 히스토리 관리 모듈, 제어 명령 전송 모듈, 결과 모니터 모듈로 구성되며 클라이언트가 관리 대상 시스템으로 터미널 연결을 할 수 있게 해준다. 클라이언트는 터미널 접속을 통해 파일 및 디렉토리 관리나 파일의 편집등과 같은 좀 더 자세한 제어를 할 수가 있다.

2.2. 중앙 제어 서버

중앙 제어 서버는 웹 서버와 연결 관리 모듈, 노

드 관리 모듈, MBean 관리 모듈, 제어 명령 관리 모듈, 제어 명령 수행 모듈로 이루어진 엔진과 RMI Client로 이루어지며 그림 3과 같다.

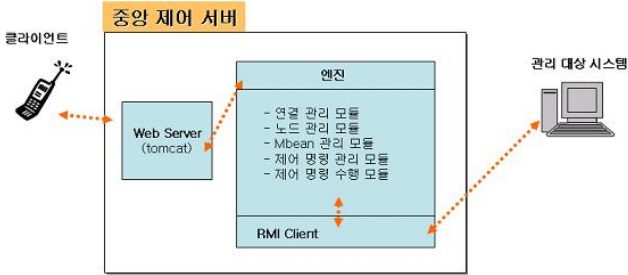


그림 3. 중앙 제어 서버 구성

클라이언트의 요청과 응답 결과는 웹 서버가 처리하며 엔진으로 명령을 전달하고 결과를 전송받아 클라이언트로 전송하게 된다. 클라이언트가 웹 서버에 접속한 후 관리 대상 시스템을 선택하면 웹 서버로부터 요청을 받은 엔진은 요청 내용을 RMI Client로 보내게 되고, RMI Client는 요청내용을 관리 대상 시스템으로 보내고 관리 요소를 전송 받게된다. 여기서 응답 받은 관리 요소들이 클라이언트가 실제 모니터링 및 제어하게 되는 항목이다. 클라이언트는 응답받은 관리 요소를 통해 제어 명령을 웹 서버를 통해 전송하면 엔진이 이를 다시 RMI Client로 보내고 RMI Client는 관리 대상 시스템에 명령을 요청하고 처리 결과를 받는다.

2.3. 관리 대상 시스템

관리 대상 시스템 구성은 그림 4와 같다. 지금까지 대부분의 분산 시스템 관리는 하나의 중앙 관리자 노드에서 폴링(polling) 기법을 이용하여 관리 대상 노드들로부터 성능 데이터를 수집하고 처리하는 중앙 집중형 관리 방식이 주로 사용되어 왔다. 그러나, 대규모 분산 시스템에서는 관리 대상 노드의 환경이 더욱 다양하기 때문에 수집된 정보의 의미적 이질성을 처리하기 위해 많은 부하가 중앙 관리 노드에 집중되는 현상이 있었다. 따라서 원격지에 있는 관리 대상 노드의 상태를 중앙 제어 서버에서 폴링하는 대신에, 에이전트를 파견하여 관리 대상 시스템에서 지역적으로 요청된 명령을 처리하고 적절한 조치를 취할 수 있게 하였다. 이런 방식으로 수집된 성능 데이터는 현재 상태를 보다 정확히 반영하며, 필요한 데이터만 수집할 수 있다.

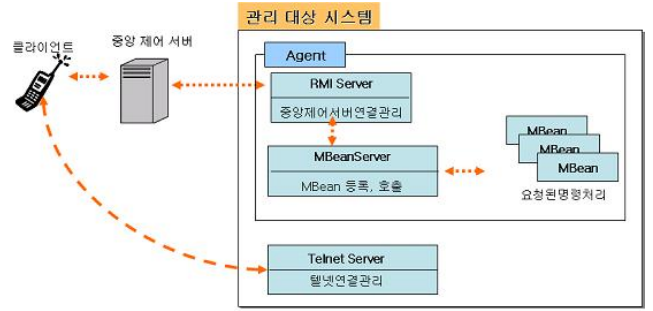


그림 4. 관리 대상 시스템 구성

Agent는 중앙 제어 서버와 통신을 하며 RMI Server, MBean Server, MBean 으로 구성된다. RMI Server는 중앙 제어 서버에서 요청한 관리 요소를 MBeanServer에 요청하고 그 처리 결과를 다시 중앙 제어 서버로 전송하는 역할을 한다. MBean Server는 MBean의 설정 파일인 MLet 에 등록된 MBean을 등록, 관리하며 RMI Server의 요청이 있을때 등록된 MBean 내용을 전송한다. MBean은 실제 요청된 명령을 처리한다. Telnet Server는 시스템에 기본 탑재된 Telnet Server를 사용하며 클라이언트의 터미널 접속을 관리한다.

3. 실행 화면

중앙 제어 서버와 관리 대상 시스템은 JMX API 1.2와 JMX Remote API 1.0을 사용하였고 플랫폼 독립적인 Java로 구현하였다. 클라이언트 단말기는 SCH-E170을 사용했으며 SK-VM 1.3 기반으로 구현하였다. 중앙 제어 서버 연결은 JMX기반으로 구현되어 MBean에 의해 운영체제, 웹 서버, 데이터베이스 서버로 관리 요소가 정의되어 관리자가 oneclick으로 시스템을 관리할 수 있게 하였다.



(a) 명령 리스트

(b) 프로세스 확인

그림 5. 프로세스 확인 화면

그림 5는 MBean에 의해 정의된 관리 요소 중 현재 실행중인 프로세스를 확인하는 `getProcess` 함수를 수행한 결과이다. 모든 명령에는 자주 사용하는 기능을 바로가기로 만들 수가 있어 초기 화면에서 자주 사용되는 명령을 바로 실행해서 결과를 얻을 수 있다. 그림 6은 바로가기 기능을 통해 현재 CPU 사용량을 확인하는 화면이다.



(a) 바로가기 저장 (b) CPU 사용량 확인
그림 6. 바로가기 실행 화면

4. 결과 분석

표 1은 기존 시스템과 제안하는 시스템을 비교 분석한 결과를 나타내고 있다.

표 1. 기존 시스템과의 비교

비교 항목	제안하는 시스템	기존 시스템[7]	기존 시스템[8]	기존 시스템[2]
사용자 단말기	휴대폰 (SCH-E170)	Simulator(Open wave WAP edition 4.0)	Simulator (Siemens S45)	PDA(Compaq iPAQ 3850)
구현언어	Java	C, Java	C	C, Java
이기종 운영체제 지원	플랫폼 독립적	플랫폼 종속적	플랫폼 종속적	플랫폼 종속적
다중 관리	가능	불가능	불가능	가능
모니터링 및 제어 항목	시스템에서 제공되는 모든 기능 사용 가능	프로세스, 사용자계정, 라우팅설정 등 제한적	휴대폰은 모니터링만 가능	사전 정의된 관리요소만 가능

본 논문에서 제안하는 분산 자원 관리를 위한 무선 원격 제어 시스템은 분산 시스템 환경에 적용하였으며 플랫폼에 독립적인 자바를 이용하여 이식성을 강화 시켰다. 또한 PDA보다 대중적인 휴대폰을 기반으로 구현하여 추가적인 장비의 구입을 최소화 하였다. 또한 중앙 제어 서버를 통해 분산된 다중 시스템을 관리할 수 있으며 MBean에 의해 사전 정의된 관리요소 이외에 보다 세부적인 제어가 필요할

경우 관리 대상 시스템에 터미널 접속을 할 수가 있어 콘솔에서 작업하듯이 시스템을 관리할 수 있다.

5. 결론

본 논문은 분산자원과 서비스를 관리하는 에이전트를 구현함으로써 모바일 단말기를 이용하여 관리자가 언제, 어디서든지 자신의 시스템을 관리할 수 있다. 그래서 본 연구의 기반이 되는 JMX 기술을 응용한 관리 모듈과 Agent와 통신하는 중앙 제어 서버를 통해 다수의 시스템을 관리할 수 있게 하였다. 그리고, 원격지 시스템에 직접 접속을 해야할 경우를 위해 모바일 단말기를 통한 원격지 터미널에 접속할 수 있는 방법도 설계 및 구현하였다.

향후 연구 과제로써 중앙 제어 서버와 관리 대상 시스템간 RMI 통신은 SSL(Secure Sockets Layer)에 기반한 RMI 소켓을 사용한 방법과 원격지 터미널 연결은 모바일 단말기에서 SSH(Secure Shell) 기반의 터미널 접속에 대한 연구가 계속 되어야 할 것이다.

참고문헌

- [1] Evan Marcus, Hal Stern, "Blueprints for High Availability", John Wiley & Sons, 2000.
- [2] 이문구, "PKI를 기반으로 한 실시간 무선 원격제어 시스템의 구현", 한국정보보호학회논문지 제 13권 3호, pp.71-79, 2003. 6.
- [3] 나승원, "무선 환경에서 자원 관리를 위한 모바일 에이전트의 설계 및 구현", 한국정보과학회 학술발표논문집(B), pp.151-153, 2003. 4.
- [4] Sun Microsystems, "Java Management Extensions Instrumentation and Agent Specifications, v1.2", October 2002.
- [5] 김진향, "JMX를 기반한 분산된 자원을 관리하는 모니터 시스템에 관한 연구", 건국대학교 석사학위, 2001. 2.
- [6] Sun Microsystems, "Java RMI Specification", <http://java.sun.com/j2se/1.4.2/docs/guide/rmi/spec/rmiTOC.html>.
- [7] 최원석, "WAP 기반 서버 관리 시스템의 설계 및 구현" 한국정보처리학회 춘계학술대회, VOL.09 NO.01, pp.1479-1482, 2002. 4.
- [8] 김찬수, "무선 인터넷 기술을 이용한 리눅스 시스템 원격관리", 호남대학교 정보통신연구 Vol.11, p.207-218, 2001.