

이동 에이전트를 이용한 원격관리 시스템 설계

홍성표^{*} · 송기범^{*} · 박찬모^{*} · 이준^{**} · 오무송^{**}

^{*}조선대학교 컴퓨터공학과

^{**}조선대학교 컴퓨터공학부

Design of Remote management System Using Mobile Agents

Seong-pyo Hong^{*} · Gi-beom Song^{*} · Chan-mo Park^{*} · Joon Lee^{**} · Moo-song Oh^{**}

^{*}Dept. of Computer Engineering, Graduate School, Chosun University

^{**}Dept. of Computer Engineering, Chosun University

요 약

분산 컴퓨팅 기술과 객체지향 기술의 발전은 인터넷상의 동일 기종간의 분산처리뿐만 아니라, 이 기종간에도 분산 컴퓨팅이 가능한 분산 객체 컴퓨팅으로 발전되었다. 이를 대표하는 CORBA(Common Object Request Broker Architecture)도 단지 네트워크를 통한 이기종간의 상호 운용성만을 해결한 상태이며, 대부분 서버 객체의 존재 및 정상적인 활동이 보장되지만 네트워크 오버헤드가 발생하는 경우에는 클라이언트의 서비스 요구에 응할 수 없는 문제점을 가지고 있다. 이러한 문제를 해결하는 방안 중 하나로, 데이터와 코드를 결합한 객체를 직접 이동하는 이동 에이전트(Mobile Agent)를 분산 컴퓨팅 기술과 결합하는 방법이 대두되고 있다.

ABSTRACT

The distributed object computing is possible to distributed computing on same or heterogeneous machine environment with growth of distributed computing and object-oriented technology. The typical of model, CORBA is just solved the interoperability in heterogeneous machine environment over the network. But, the CORBA has a problem that the server is can't rely on request of client occurs network overhead by extinction of network. A plan was on the rise to solve the problem that is combination of mobile agent and distributed computing technology.

1. 서 론

분산 컴퓨팅 기술과 객체지향 기술의 발전은 인터넷상의 동일 기종간의 분산처리뿐만 아니라, 이 기종간에도 분산 컴퓨팅이 가능한 분산 객체 컴퓨팅으로 발전되었다. 이를 대표하는 OMG(Object management Group)의 CORBA(Common Object Request Broker Architecture)명세에 따른 분산 객체 컴퓨팅은 실행 단위를 객체로 분리하고, 객체간의 통신을 메시지 전달을 통해 객체간 상호 운용성(Interoperability)을 보장하고 있다.

그러나, CORBA에서 제안한 분산 객체 컴퓨팅도 단지 네트워크를 통한 이 기종간의 상호 운

용성을 해결한 상태이며 기존의 분산 컴퓨팅처럼 다른 노드에 서버 역할을 하는 코드, 즉 서버 객체가 존재하는 형태로 되어있다. 대부분은 서버 객체의 존재 및 정상적인 활동이 보장되지만 네트워크 단절이라든가 수많은 클라이언트의 서비스 요구에 의해서 네트워크 오버헤드가 발생하는 경우에는 클라이언트의 서비스 요구에 응할 수 없는 문제를 일으킨다. 또한 데이터 전송 및 접속 유지에 따른 네트워크 오버헤드를 해결하지 못하고 있다.[1][4][5]

이러한 문제를 해결하는 방안 중 하나로서, 데이터와 코드를 결합한 객체의 이동 즉, 이동 에이전트(Mobile Agent)를 사용하는 방안이 제안되었다. 이동 에이전트는 네트워크 트래픽의 감소, 비

동기 통신 등의 여러 가지 이점을 제공하고 있으며, 전자 상거래 서비스, 워크플로우 관리 시스템, 네트워크 관리 등에 활용되고 있다.[2][3][6]

본 논문에서는 Java와 일본 IBM 연구소에서 개발한 Java 기반 자율 이동에이전트 시스템인 Aglet Workbench를 이용하여 원격관리 시스템을 설계하였다. 원격관리 시스템은 이동에이전트를 이용하여 여러 호스트들을 이동하면서 정보를 수집하여 마스터로 전송하는 시스템이다.

II. 에이전트

에이전트(Agent)란 용어에 대한 일반적인 정의는 '사용자가 원하는 작업을 사용자를 대신하여 해결하여 주는 소프트웨어' 또는 '특정 목적을 수행하기 위하여 사용자를 대신하여 작업을 수행하는 자율적인 프로세스(Autonomous Process)'라고 할 수 있다.

에이전트의 정의를 살펴보면 기본적으로 에이전트라고 불리기 위해서 필요한 몇 가지 특성을 추출해낼 수 있는데 대표적으로 자율성(autonomous), 지능(intelligence), 이동성(mobility), 사교성(social ability)등을 들 수 있다.

특히, 이동성은 그림. 1과 같이 기존의 클라이언트/서버의 개념과는 판이하게 다른 개념으로 서버의 내용을 클라이언트를 통해 전송 받아 정보를 얻거나 작업 수행을 하는 것이 아니고 클라이언트가 필요로 하는 작업을 위해 에이전트를 서버로 보내어 수행을 시킨다. 이렇게 함으로써 네트워크의 통신부하를 줄일 수 있을 뿐만 아니라 네트워크의 지연(latency)현상을 극복할 수 있다.[3][4][6]

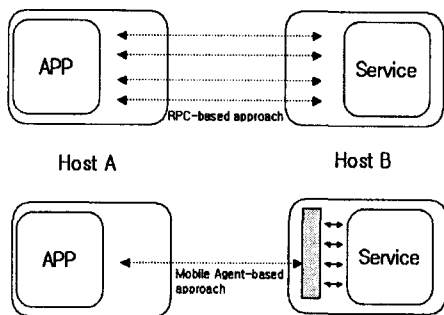


그림. 1 Client/server paradigm과 Mobile agent paradigm 비교

이러한 에이전트 기술은 세 가지 주요한 요소로 나눌 수 있다.

① 에이전트 프로그래밍 언어: 에이전트와 에이전트가 방문할 Place를 개발하기 위한 언어

② 에이전트 시스템: 에이전트를 위한 가상 기계(virtual machine) 제공

③ 에이전트 프로토콜: 에이전트간의 통신을 위한 규약

에이전트가 이질적인 망(Heterogeneous Network)의 호스트 사이를 이동하며 수행되기 위해서는 코드뿐만 아니라 상태도 이동되어야 한다. 코드는 이동될 모든 호스트에 똑같은 형식으로 동작되어야하기 때문에, 직접 해석(Interpretation)되거나 다시 컴파일하지 않고 수행될 수 있는 이식성 있는 중간 언어(portable intermediate interpreter-based language)가 되어야 한다. 따라서, 에이전트 언어는 이와 같은 요구사항을 충족하면서도 효율적으로 프로그래밍 할 수 있어야 하기 때문에 현재 개발된 대부분의 언어는 해석 언어(Interpreter language)의 형태를 가지고 있으며, 객체 지향 기법을 사용함으로써 플랫폼에 독립적인 에이전트의 개발을 가능하게 한다.

대표적인 에이전트 개발 언어로는 Java, Tcl/Tk(Tool Command Language/Toolkit), Telescript 등이 있다. Tcl/Tk는 비교적 간단한 스크립트 언어로써, 여러 플랫폼에서 가용성(Availability)을 가지며 GUI의 생성을 지원하는 API로 Tk를 지원한다.

이중에서 Java는 'better C++'라고 불릴 만큼 C++과 유사한 객체지향 언어이면서, 인터넷 언어로서 네트워크 환경에 유용한 언어로, 이동형 에이전트의 프로그래밍에 매우 적합한 언어로 인식되고 있다.[3][5][7][8]

에이전트 시스템은 에이전트를 생성, 해석, 수행, 폐기 등의 생명 주기를 관리하는 플랫폼이다. 이동 에이전트 시스템의 개념적인 모델은 에이전트와 Place라고 하는 두 가지 핵심 개념에 기초를 두고있다. 이 개념을 기반으로 에이전트의 행위(Behavior)와 네트워크 상의 이동 및 에이전트간의 통신이 이루어진다.[4][5][8][9]

III. 시스템 설계 및 구현

본 논문에서 원격관리 시스템을 구현하는데 Java와 IBM Aglets Workbench를 이용하였다.

이동에이전트(Mobile Agent)는 코드뿐만 아니라 상태도 이질적인 망(Heterogeneous Network)의 호스트 사이를 이동하며 수행된다. 그렇기 때문에 에이전트 프로그래밍 언어는 직접 해석(Interpretation)되거나 다시 컴파일하지 않고 수행될 수 있는 이식성 있는 중간 언어(portable intermediate interpreter-based language)가 되어야 하는데 Java는 이러한 조건을 잘 만족한다.

IBM Aglet Workbench는 IBM에서 개발중인 Java에 근거한 자율적인 이동에이전트 시스템으로 시각적인 개발환경을 지원하기 때문에 에이전트 프로그램 작성과 실행이 다른 환경에 비해 용

이하고, 네트워크 상을 이동하는 플랫폼에 독립적인 에이전트의 작성을 가능하게 해준다.

3.1 시스템 구조

이동에이전트를 이용한 원격관리시스템은 기존의 클라이언트/서버 모델의 시스템과는 달리 관리대상이 되는 시스템에 클라이언트 프로그램을 직접 설치하지 않는다. 대신에, 에이전트를 생성하여 대상시스템에 파견한 후 실행하여 정보를 수집한다.

원격관리시스템은 두 개의 에이전트로 이루어져 있다.

- Master Agent
- Slave Agent

Master 에이전트는 고정에이전트로, 사용자에게 수집한 정보를 보여주기 위한 사용자 인터페이스, 관리대상이 되는 원격시스템으로 파견될 에이전트를 생성하는 클래스, 파견된 에이전트가 전송해온 정보를 수신하는 클래스, 수집된 정보를 분석, 가공하는 클래스로 이루어져 있다.

Slave 에이전트는 이동에이전트로, 대상시스템의 정보를 수집하는 클래스와 수집된 정보를 Master 에이전트로 전송하는 클래스로 구성되어 있다.

원격관리 시스템의 동작과정은 그림. 2 와 같다.

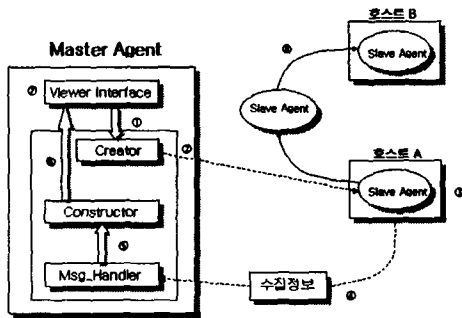


그림. 2 시스템 구조

- ① 사용자가 Viewer를 통해 검색대상이 되는 호스트를 결정한다.
- ② Creator 클래스는 전달받은 호스트 목록별로 Slave agent를 생성하고 대상호스트로 파견한다.
- ③ 파견된 Slave agent는 대상호스트에 도착한 후 시스템 정보, 계정정보 등을 수집한다.
- ④ 각 Slave 에이전트는 마스터의 MsgHandler 클래스에게 수집한 정보 내용을 보고한다.
- ⑤ MsgHandler 클래스는 보고 내용을 Construct-

or 클래스로 전달한다.

- ⑥ Constructor 클래스는 전달받은 정보를 분석하고 가공한 후 분석 내용을 Viewer로 전달한다.
- ⑦ Viewer는 전달받은 결과를 화면에 표시한다.
- ⑧ Slave 에이전트는 점검 대상의 마지막 목적지 호스트에 도착할 때까지 연속적으로 이동하며 정보 수집을 반복한다.

이동에이전트에 의해서 수집되는 정보는 다음과 같다.

- 시스템 정보: 호스트 이름, 운영체제, 네트워크 정보, 원격시스템의 시간 등 시스템으로부터 기본적으로 얻을 수 있는 정보
- 계정 정보: 전체 사용자의 수 및 각각의 사용자에 대해 시스템으로부터 얻을 수 있는 정보

3.2 구현

이동에이전트를 이용한 원격관리시스템은 Java (JDK1.1.8)와 IBM Aglets Workbench를 이용하여 구현하였다. Master Agent가 실행되는 Master 시스템과 Slave Agent가 실행될 관리대상시스템으로는 Windows2000이 탑재된 Pc를 사용하였다.

그림. 3은 이동에이전트가 관리대상 호스트로 이동하여 수집한 정보를 Master 시스템에서 출력한 것이다.

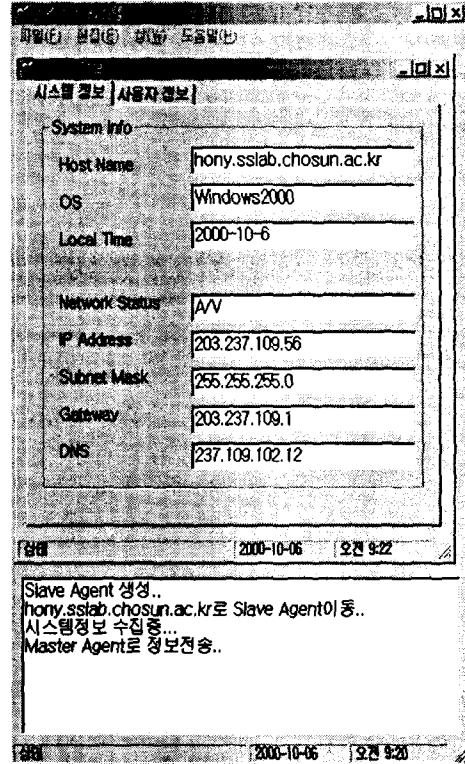


그림. 3 원격관리 시스템 주화면

이동 에이전트를 이용한 원격관리시스템은 기존의 클라이언트/서버 환경에서 구현된 기능을 이동 에이전트 환경으로 구현한 것으로, 이동 에이전트를 이용하여 원격시스템 관리가 가능함을 보여주고 있다.

향후 이동 에이전트간의 통신기능을 추가한다면 이동 에이전트를 이용한 어플리케이션에서 더 많은 장점을 얻을 것으로 생각된다.

IV. 결론

기존의 분산 컴퓨팅 시스템인 클라이언트/서버 시스템은 이기종 환경에서 시스템운영상의 어려움과 데이터 전송 및 접속 유지에 따른 네트워크의 통신 부하 문제, 지연 현상 문제를 가지고 있다. 이를 해결하기 위해 이기종간에도 컴퓨팅이 가능한 분산 객체 컴퓨팅이 개발되었는데, 이를 대표하는 것으로는 CORBA가 있다.

그러나, CORBA도 단지 네트워크를 통한 이기종간의 상호 운용성만을 해결한 상태이다.

이러한 통신 부하 및 지연현상을 해결하기 위한 방법으로 데이터와 코드를 결합한 객체를 직접 이동하는 이동 에이전트 개념이 제안되었다. 이동 에이전트는 객체의 이동시와 결과 값의 전송시에만 네트워크를 이용하기 때문에 네트워크 부하를 줄임으로써, 시스템의 자원절약이 가능하다.

본 논문에서 설계하고 구현된 이동 에이전트를 이용한 원격관리 시스템은 각각의 기능별로 에이전트 구성이 가능하기 때문에 새로운 기능의 추가나 삭제 등이 용이한 장점을 가진다.

향후 연구 과제로는 각 호스트를 이동중인 이동 에이전트간에 서로 통신을 통하여 협력할 수 있는 기능을 위한 연구 등이 있다.

참고문헌

- [1] OBJECT MANAGEMENT GROUP. The Common Object Request Broker: Architecture and Specification, 2.0 ed., July 1995.
- [2] Vu Anh Pham and Ahmed Karmouch, "Mobile Software Agents : An Overview, " IEEE Comm. Magazine. pp.26-27, July 1988
- [3] H. S. Nwana and M. Wooldridge, "Software agent technologies, " BT Technol. J, pp.68-78, Oct. 196
- [4] D.Wong, N. Paciorek, and D. Moore, "Java-based Mobile Agents," CACM, pp.92-102, Mar 1999
- [5] G, Cabri, L. Leonardi, and F. Zambonelli, "A Case Study in Mobile Agent Coordination, " PDPTA '98 Int'l Conf., pp.810-817, 1998
- [6] Robert L. Popp, Bohdan P. Maksymiuk and Michael R. Poreda, "Efficient Information

Re-trieval On the World Wide Web Using Adapt-able And Mobile Java Agents," IEEE '98 Int'l conf., pp.2219-2224, 1998

[7] Benzamin Falchuk and Ahmed Karmouch, "The Mobile Agent Paradigm Meets Digital Document Technology: Designing for Automous Media Collection," Multimedia Tools and Appl., pp. 153-162, 1999

[8] D.B Lange and D.T. Chang, "IBM Aglets Workbench : Programming Mobile Agents in Java," IBM White paper, <http://www.trl.ibm.co.jp/aglets/whitepaper.html>, Sep.1996

[9] H.Peine and T.Stolpman, "The Architecture of the Ara Platform for Mobile Agents, " Pro-ceedings of the First Int.l Workshop on Mobile Agents, p. 12, Apr. 1997