

자바 플랫폼 지니 기반 소형 자바가상기계의 설계 및 구현

정 준 영*, 최 원 호*, 김 도 우*, 조 증 보*, 유 흥 식*, 정 민 수*

*경남대학교 컴퓨터공학과,

Design and Implementation of small sized

Java Virtual Machine on Java Platform Jini

Jun-Young Jung*, Won-Ho Choi*, Do-Woo Kim*, Jeung-Bo Cho*, Hong-Sik Yoo*, Min-Soo Jung*

*Dept. of Computer Engineering, Kyungnam University,

요 약

모바일 인터넷 시대가 열리면서 언제, 어디서든지 네트워크에 연결될 수 있는 휴대용 디지털 기기의 인터넷에 대한 관심이 증대되고 있다. 모바일 단말기에서의 자바 플랫폼 기반 지니 홈네트워크 접속은 자바 기술의 특성을 가장 잘 살린 컴퓨팅 환경이다. 지니를 지원하기 위해서는 자바 가상 기계라는 소프트웨어가 실행을 하고 있는 환경이어야 하고 RMI 를 지원해야 하는데 Java2ME 의 K-자바 가상 기계는 이를 지원하지 못한다.

본 논문에서는 Java2ME CLDC 를 기반으로 지니에 접속할 수 있는 자바 가상 기계에 대해 논의 하였다. 지니 KVM 은 각 모바일 단말기에 맞는 프로그램을 하지 않아도 되는 환경을 제공할 뿐만 아니라, 표준 자바 클래스 파일 포맷과 CLDC 클래스 파일을 지원하고, 지니에 접속할 수 환경을 제공함으로써, 모바일 단말기의 본격적인 네트워킹을 할 수 있도록 한다.

1. 서 론

오늘날의 컴퓨팅 시장은 데스크탑에서의 인터넷 시대를 지나 언제, 어디서든지 네트워크에 연결될 수 있는 휴대용 디지털 기기의 인터넷에 대한 관심이 증대되고 있다. 이들 반영하듯 이동성과 휴대성을 모태로 주목 받고 있는 디지털기기 가운데 하나가 셀룰러 폰, PDA등과 같은 이동 단말기들이며, 기존의 컴퓨팅 시장에서 큰 변혁을 초래하고 있다. 이러한 변혁을 주도할 기술로, 자바와 지니가 대표적이다. 자바는 동적인 애플리케이션 다운로드, 크로스 플랫폼 호환성, 향상된 사용자 경험과 역동성, 비연결성, 보안 문제등과 같은 이유로 각광 받고 있으며, 지니는 자바의 특성을 이용환경 전체로 확장하여 '네트워크 상

의 모든 요소를 연계 시키는 환경을 실현하기 위해 등장한 기술' 이다. 하지만 데스크 탑 플랫폼에서 수행되던 자바 기술을 셀룰러폰이나 PDA와 같은 휴대용 디지털 기기에 그대로 적용할 수는 없다. 왜냐하면, 기존의 컴퓨팅 환경에서 사용되어 오던 자바는 많은 자원을 필요로 하는 반면, 셀룰러폰이나 PDA와 같은 제한적인 자원을 가지고 있는 기기에서는 자바의 모든 기술을 받아들일 수가 없기 때문이다.

썬마이크로시스템즈사는 K-자바 가상 기계라는 제품을 내놓게 되었다. 적은 메모리의 용량과 CPU 처리능력 그리고 느린 무선 네트워크에 맞게 설계된 가상 기계로, 자바 원격 메소드 호출(Remote Method Invocation : RMI) 기능을 제거하여 지니와의 접속이 이루어지지 않는다

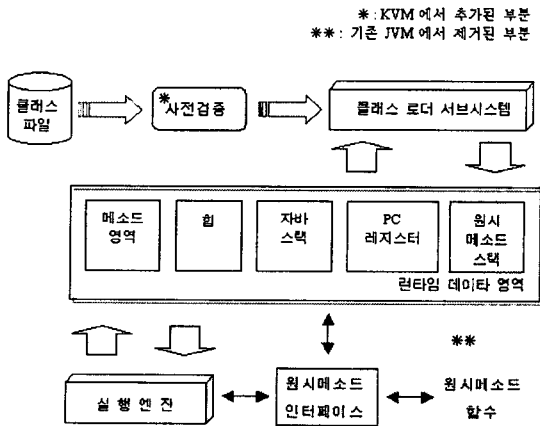
† 정보통신부에서 지원하는 2000 대학기초연구지원 사업으로 수행

본 논문의 구성은 다음과 같다. 2장에서는 소형 자바 가상기계의 구조와 실행과정을 살펴보고, 3장에서는 지니에 접속할 수 있는 소형 자바 가상 기계의 구성과 역할에 대해 기술하고, 4장에서는 지니에 접속할 수 있는 소형 자바 가상 기계의 구현 논점을 기술한다. 그리고 5장에서는 결론 및 활용 방향을 제시한다.

2. 관련 연구

2.1 K-자바 가상 기계의 구조

K-자바 가상 기계의 구조는 <그림 2-1>와 같이 클래스 로더 시스템(Class Loader System), 실행 엔진(Execution Engine), 런타임 데이터 영역(RunTime Data Areas)으로 구성되어 있다. 기본적으로 기존의 자바 가상 기계를 유사하다.



<그림2-1> K-자바가상기계의 내부구조

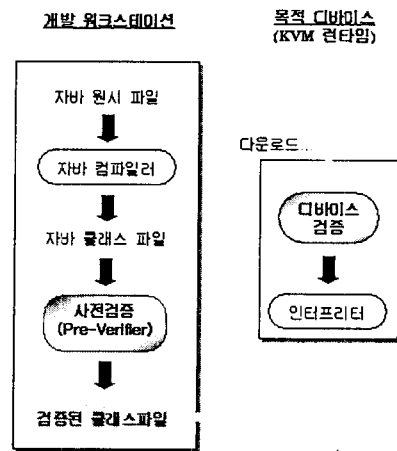
하지만 K-자바 가상 기계를 클래스 파일을 클래스 로더 서버 시스템에 로드하기 전에 사전검증을 거쳐 미리 검증한다. 그리고 K-자바 가상 기계를 런타임 데이터영역에는 원시 메소드 함수를 지원하기 위한 원시 메소드 스택을 가지고 있지 않다.

2.2 K-자바가상기계 검증

가상기계에서의 검증(Verification)은 클래스 파일을

를 적재(로드 : Load)하기 위한 클래스 파일 로더 시스템에서 실행된다.

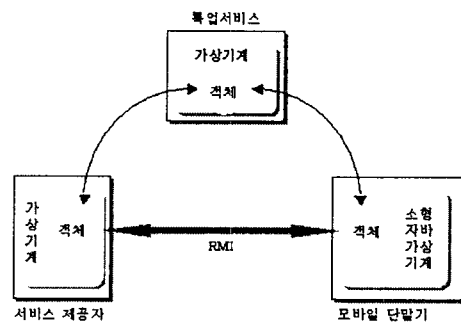
검증은 적재된 데이터의 형이 자바 언어의 의미(semantics)를 따르고 있는지 혹은 가상 기계의 내부 형식을 만족하는지를 검사하는 단계이다. 자바 가상 기계와 K-자바 가상 기계를 기본적으로 동일한 실행과정을 거치지만 실행방법에서는 다소 차이를 보인다



<그림2-2> K-자바가상기계 검증단계

3. 지니에 접속 가능한 소형 자바 가상 기계

3.1 지니와 소형 자바 가상 기계를 적용한 시스템 구성



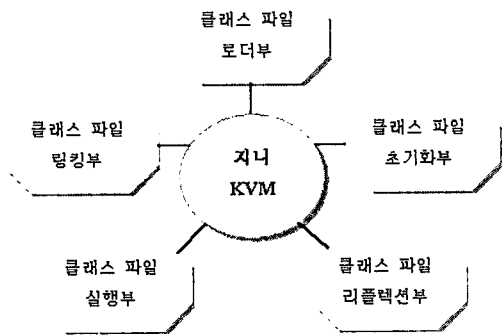
<그림3-1> 지니KVM 시스템 구성

자바 기반 분산 시스템 아키텍처중의 하나인 지니는 서비스와 클라이언트, 특업서비스로 구성되어 있

다. 본 논문에서의 시스템 구성은 <그림 1>과 같이 클라이언트로써 지니 시스템의 구성원으로 참여할 모바일 디바이스는 지니KVM을 내장하고 지니 시스템과의 상호작용을 위해 RMI를 지원해야 하는 지니 KVM이다.

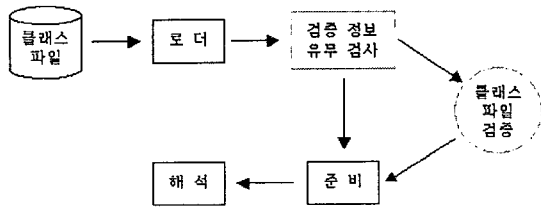
3.2 지니KVM의 구성요소

지니 접속을 지원하기 위한 소형 자바 가상 기계의 구성은 <그림 3-2>에서 보는 바와 같이 로더 단계, 초기화 단계, 실행 단계의 소스 레벨 구성은 Java2ME CLDC의 K-가상기계 소스레벨의 구성과 동일하다.



<그림3-2> 지니KVM 구성요소

3.3 배치단계에서의 검증



<그림3-3> 배치단계에서의 검증

<그림3>에서 보는 바와 같이 로드되는 클래스 파일의 검증 정보 유무를 체크 하여 검증 정보가 포함되어 있으면 Java2ME명세서를 기반으로 하는 검증단계를 수행하고, 검증 정보가 포

함되어 있지 않으면, Java2ME CLDC의 클래스 파일 포맷이 아니므로 검증 단계를 생략한다.

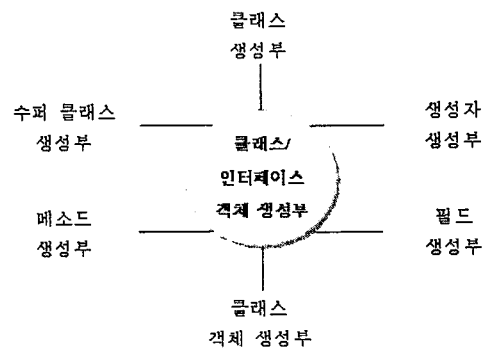
3.4 클래스 파일 리플렉션

자바 프로그램이 실행 시간에 객체의 멤버나 클래스에 대한 정보를 얻고, 이를 이용하여 원격 메소드를 호출하여 클라이언트에서의 지니 KVM에서 실행할 객체를 생성하여 클래스 멤버들에 대한 작업을 할 수 있도록 한다.



<그림3-4> 리플렉션 구성

리플렉션은 <그림3-4>에서 보는 바와 같이 지니에서 사용되는 록업 서비스에서 전송 받은 데이터를 먼저 분석하는 메시지 분석부와 분석한 메시지를 바탕으로 클라이언트에서 생성하게 될 클래스/인터페이스 객체 생성부로 구성된다. 클래스/인터페이스 객체 생성부에서는 서비스의 객체와 서비스의 클래스 메소드를 호출할 수 있는 객체를 만든다.



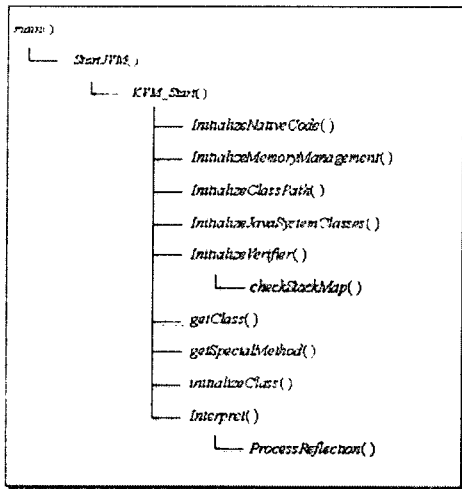
<그림3-5> 클래스/인터페이스 객체생성부

<그림3-5>과 같이 서버의 객체에 대해 저장되어 있는 객체의 클래스에 관한 정보, 인터페이스에 관한 정보, 생성자에 관한 정보, 필드에 관한 정보, 메소드

에 관한 정보를 얻으면, 클라이언트의 소형 자바 가상 기계는 객체를 생성하여 서버의 자바 가상 기계에서 실행되고 있는 객체의 메소드를 호출이 이루어진다.

4. 구현 논점

4.1 지니KVM의 주요 모듈



<그림4-1>지니KVM의 주요모듈리스트

자바 가상 기계를 직접 구현하지는 않고 소스가 공개된 썬마이크로시스템즈사의 Java2ME CLDC 에 포함된 KVM을 기반으로 구현하였다. 자바 가상 기계의 사양에 나타나 있는 것처럼 KVM 실행순서는 표준 자바 프로그램의 실행 순서를 따르고 있다.

- InitializeClassPath() : 클래스 로딩을 위한 클래스 스페스를 초기화한다.
- InitializeThreading() : 첫번째 실행 쓰레드를 생성하고 가상 기계 레지스터를 초기화한다.
- InitializeJavaSystemClasses() : 자바 가상 기계가 필요로 하는 자바 클래스들을 로드하고 초기화 한다.
- InitializeVerifier() : 바이트코드 검증을 초기화 한다.

- checkStackMap() : 클래스 파일을 로드한 후 검증 정보의 유무를 클래스 파일을 통해 검사하고, 있으면 검증을 시작하고 없으면 검증을 실행하지 않는다.
- getClass() : 주어진 이름을 가진 클래스를 찾아서 필요하다면 클래스를 시스템에 로드한다.
- getSpecialMethod() : 간단한 선형 록업방법을 사용해서 <clinit>, main과 같이 명시된 특정 메소드를 찾는다.
- initializeClass() : 클래스(java.lang.System, java.lang.String, java.lang.Thread, java.lang.Class)를 로드하고 난 후에 internal static constructor <clinit>을 실행함으로써 초기화 될 수 있다. 이 함수는 필요한 Class.runClinit 프레임을 설정하고 인터프리터에게 반환한다.
- Interpret() : 실제 바이트코드를 해석하고 실행하는 부분으로 지니KVM에서는 아주 중요한 루틴이다.

5. 결 론

본 논문에서 설계한 지니KVM은 클래스 파일의 포맷을 체크하여 실행하고, Java2ME CLDC 클래스 파일 포맷과 Java2SE의 표준 자바 클래스 파일 포맷의 실행을 지원함으로써 Java2 플랫폼 사이의 호환이 가능하다. 그리고 소형 자바 가상 기계에 리플렉션 기능을 추가함으로써 자바 기반 지니 홈 네트워킹과의 상호 작용을 위해 필요한 RMI기능을 부여하여 소형 자바 가상 기계를 탑재한 모바일 단말기 및 디지털 기기는 인위적인 조작이나 설치와 같은 절차 없이 네트워크에 연결되지만 하면 시간과 장소에 상관없이 네트워크 상에서 서로 간의 상호작용을 통해 여러 가지의 작업이 가능하다.

모바일 단말기나 디지털 기기에 프로그래밍 되어 있는 기계어나 C언어를 자바 언어로 프로그래밍 함으로써 각 모바일 기기에 맞게 프로그래밍 할 필요

가 없는 환경으로 활용할 수 있고, 지니와 디지털 기기 사이의 자바 가상 기계를 탑재하기 위한 최적화 환경을 제공하기 위한 자료로서도 활용이 가능하다.

참 고 문 헌

- [1]B. Venners, *Inside Java Virtual Machine*, McGraw-Hill, (1997)
- [2]W. KEITH EDWARDS, *Core Jini*, Sun MicorSystems(1999)
- [3]Sing Li, *Professional Jini*, Wrox(2000)
- [4]옥재호, “비주얼 자바 클래스 파일 브라우저의 설계 및 구현”, 경남대학교 대학원 석사논문, (1998)
- [5]김도우, “비주얼 바이트코드 시뮬레이터의 설계 및 구현”, 경남대학교 대학원 석사논문, (1998)
- [6]이종동, “자바 가상 기계 프로파일러의 설계 및 구현”, 경남대학교 대학원 석사논문, (1998)
- [7]<http://java.sun.com/>, Sun Microsystems, Java Home Page
- [8]<http://artima.com/>, a resource for Java and Jini developers
- [9]<http://www.jini.org/>, Jini Home Page
- [11]<http://www.jiniworld.net/main.html>, Jini and HomeNetwork
- [12]<http://java.sun.com/j2me/index.html>, Java2 Platform Micro Edition Specification, White Paper
- [13]<http://java.sun.com/j2se/1.3/docs/guide/rmi/>, RMI Specification
- [14]<http://java.sun.com/products/jdk/1.1/docs/guide/reflection/spec/java-reflectionTOC.doc.html>, Java Core Reflection