모바일 클라우드 디바이스 개발을 위한 안드로이드 SPICE 프로토콜 연구

정준권*, 정성민*, 김남욱*, 정태명**
*성균관대학교 전자전기컴퓨터공학과
**성균관대학교 정보통신대학
e-mail:*{jkjung, smjung, nukim}@imtl.skku.ac.kr
**tmchung@ece.skku.ac.kr

A Study on Android SPICE Protocol for Development of Mobile Cloud Device

Jun-Kwon Jung*, Sung-Min Jung*, Nam-Uk Kim*, Tai-Myoung Chung**
*Department of Electrical and Computer Engineering,
Sungkyunkwan University
**College of Information and Communication Engineering,
Sungkyunkwan University

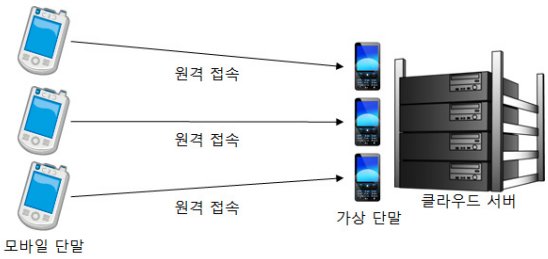
요 약

스마트폰과 같은 모바일 단말은 모바일 클라우드의 서비스를 받기 위해 모바일 가상머신에 원격접속할 수 있는 기술이 필요하다. 원격접속 표준 프로토콜은 프레임 버퍼 방식의 Remote Frame Buffer(RFB) 프로토콜과 메시지 전송방식의 Remote Desktop Protocol(RDP)로 나뉜다. SPICE 프로토콜은 RDP 프로토콜 기반의 오픈소스 원격접속 프로토콜이다. SPICE 프로토콜은 가상머신에 원격접속 할 수 있지만 모바일 환경을 지원하지 않는다. 따라서 SPICE 프로토콜을 모바일 환경에서 사용하려면 포팅 작업이 필요하다. 본 논문은 리눅스용 SPICE 클라이언트 모듈을 안드로이드 환경에 맞도록 포팅하는 방법을 소개하고자 한다. 또한 포팅시 발생하는 문제점과 이를 해결하는 방법 몇 가지를 함께 소개한다.

1. 서론

클라우드 컴퓨팅 기술이 발전하면서 다양한 플랫폼을 통한 클라우드 서비스의 지원이 가능해졌다. 스마트폰이 확산되면서 이를 통한 컴퓨팅 서비스에 대한 요구가 늘어났고 모바일 클라우드의 요구가 생기기 시작했다. 모바일 클라우드 서비스란 모바일 단말이 클라우드 서버에서 제공하는 가상단말에 원격접속하여 보유한 단말보다 고성능의 모바일 환경을 제공받는 서비스를 의미한다. 이러한 모바일 클라우드는 플랫폼에 독립적인 서비스를 지원할 필요성도 있다. 클라우드 사용자는 모바일 가상머신에 원격접속하여 원하는 서비스를 받을 수 있다. 원격접속을 통한 통일된 서비스 플랫폼이 생성되면 서비스 개발자들은 여러 플랫폼에 대한 호환성을 고려하지 않아도 되므로 더 효율적이고 더 높은 수준의 어플리케이션을 개발하고 서비스 할 수 있게 된다. (그림 1)은 모바일 클라우드 서비스의 간단한 형태를 보여준다.

이러한 가상머신에 접속하기 위해선 가상머신 환경에 맞는 원격접속 서비스가 필요하다. SPICE 프로토콜은 QEMU라는 가상화 프로그램에 최적화된 원격접속 프로토콜이다[1]. 본 논문은 SPICE 클라이언트 프로그램을 모바일 클라우드 환경에 적합한 형태로 변환하는 것을 소개한다. 2장은 원격접속 프로토콜이 무엇인지를 소개하고 대표적인 두 가지 프로토콜의 특징을 비교한다. 3장은 SPICE 프로토콜의 기본적인 구조를 살펴본다. 4장은 SPICE 클라이언트 프로그램을 모바일 클라이언트 환경에 맞는 포팅에 대한 내용을 기술하며, 5장에서 향후 연구에 관한 내용을 설명한다.

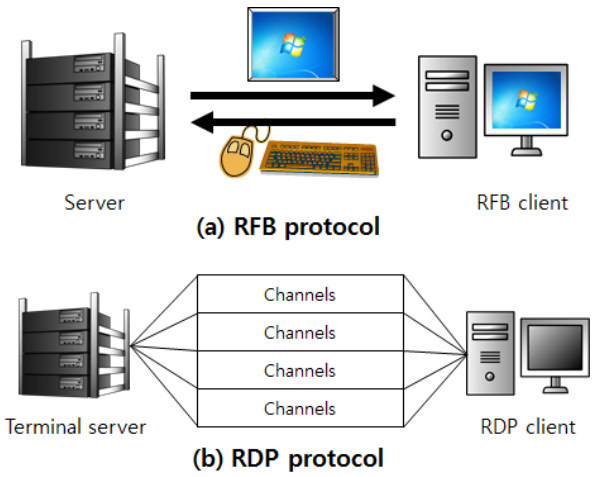


(그림 1) 모바일 클라우드 서비스

2. GUI형 원격접속 프로토콜

GUI형 원격접속 프로토콜은 크게 RFB 프로토콜과 RDP 프로토콜로 나눌 수 있다[2]. RFB 프로토콜은 원격 컴퓨터의 출력화면을 캡처하여 클라이언트에 전송하는 방식을 사용한다. 프레임 버퍼를 활용하여 원격 컴퓨터의 화면 그대로를 클라이언트에 출력하여 보여주는 방식으로 구현되어 있다[3]. RFB 프로토콜은 구조가 간단하기 때문

에 다양한 플랫폼에 활용되고 있고, 주로 Virtual Network Computing(VNC)라는 이름의 프로그램으로 사용된다. VNC는 소스코드가 공개되어 있어 RealVNC와 같은 몇몇 상용프로그램을 제외하고 대부분은 무료버전으로 사용자들에게 제공된다. RDP 프로토콜은 원격 컴퓨터의 디바이스의 메시지를 전송받아 클라이언트에서 메시지를 처리하여 화면에 나타나는 방식을 활용한다. 이때 각각의 메시지 종류별로 채널을 구성할 수 있고, 최대 64,000개의 채널을 구성할 수 있도록 지원한다[4]. 마이크로소프트사가 RDP 프로토콜의 라이선스를 보유하고 있어, 윈도우 환경을 제외한 서버 플랫폼은 지원되지 않는다[5]. 다만, 클라이언트는 다양한 OS플랫폼을 지원한다. (그림 2)는 RFB 프로토콜과 RDP 프로토콜의 기본적인 구조를 나타낸다.

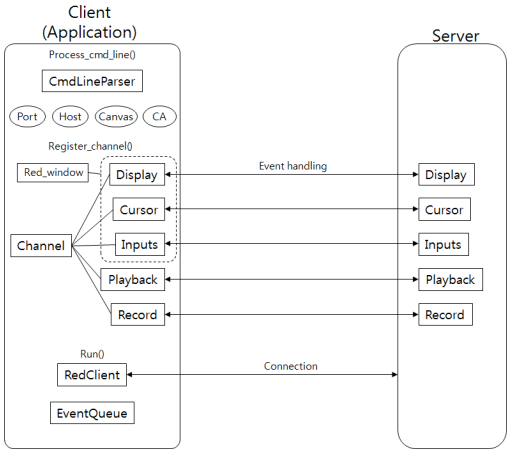


(그림 2) RFB와 RDP 프로토콜

3. SPICE 프로토콜

SPICE 프로토콜은 RDP 프로토콜과 유사한 원격접속 프로토콜이다. RDP 프로토콜은 라이선스 문제로 인해 윈도우에 종속적인 특징을 가지고 있는 반면에 SPICE 프로토콜은 QEMU라는 가상머신 에뮬레이터에 종속된 오픈소스 프로토콜이다. RDP 프로토콜과 마찬가지로 여러 개의 채널을 만들어 각각의 채널마다 입력, 영상, 음성 등의 데이터를 개별적으로 전송, 수신한다. SPICE 서버는 QEMU와 연동되어 동작하므로 QEMU를 지원하는 모든 플랫폼에서 동작한다. 그리고 SPICE 클라이언트는 리눅스, 윈도우, 맥OS 환경을 지원한다. (그림 3)은 SPICE 클라이언트의 기본적인 동작 구조를 보여준다. 기본 구조는 C로 작성이 되어 있고, 플랫폼 환경에 따라 유저 인터페이스는 다양한 형태로 제작되어 있다.

모바일 클라우드 서비스는 모바일 단말들에게 고성능의 컴퓨팅 환경을 제공한다. 클라우드 서버는 각각의 클라이언트에게 하나의 모바일 가상단말을 생성하여 연결하게 된다. 이 때, 사용자의 모바일 단말과 클라우드 가상단말의 연결을 원격접속 프로토콜로 수행하게 된다. 모바일 환경은 일반적인 PC 환경에 비해 낮은 컴퓨팅 성능과 네트워크 속도를 보장한다. 그러므로 모바일 환경에 맞는 원격접속 프로토콜 또한 오고가는 트래픽이 적어야 하며 원격 서버 화면을 출력하는데 부하가 적거나 없어야 한다. 따라서 RFB 프로토콜은 모바일 환경에 적합하지 않다. RDP 프로토콜은 구조상 적은 트래픽이 발생되지만 라이선스 문제 상 윈도우 환경밖에는 활용할 수 없다. SPICE 프로토콜은 RDP의 라이선스 문제가 없으며 발생되는 트래픽 또한 적은 편이다. 그리고 SPICE 프로토콜은 가상화 환경에서 동작하도록 설계되어 있어 가상화 환경에서 최적의 성능을 발휘할 수 있다. 그러므로 모바일 클라우드 환경에서는 SPICE 프로토콜이 가장 적합한 원격접속이 될 것이다 하지만, SPICE 프로토콜은 아직 어떠한 모바일 환경도 공식적으로 지원하지 않는다. 따라서 본 논문은 안드로이드 환경에서 SPICE 프로토콜을 활용한 원격접속 방식을 제시한다.



(그림 3) SPICE 클라이언트

4. 안드로이드용 SPICE 클라이언트

SPICE 프로토콜은 안드로이드 버전을 지원하지 않는다. 따라서 SPICE 프로토콜을 안드로이드 환경에 사용하게 하기 위해선 리눅스용 SPICE 프로토콜을 포팅하여 사용해야 한다. SPICE 프로토콜을 안드로이드용으로 포팅하기 위해선 소스코드를 일부 수정해야 한다. 포팅 대상은 SPICE 클라이언트이다. SPICE 서버는 클라우드 서버의 QEMU에 포함되어 있고, 가상머신 내부의 운영체제에 영향을 받지 않으므로 포팅 대상에서 제외된다. 포팅할 SPICE 클라이언트는 다양한 리눅스 모듈을 활용하는데 이 모듈들 또한 안드로이드 환경에 맞도록 포팅해야 한다. SPICE 클라이언트 포팅을 위한 준비환경은 아래와 같다.

- Ubuntu Linux 11.10 32-bit
- Android SDK-2.3.3, NDK-r4c
- iconv-1.13.1, gettext-0.18.1.1, glib-2.28.1, pixman-0.20.0, jpeg-6b, openssl-1.0.0
- androidSpice 소스코드

구글코드 사이트에 개인이 SPICE 소스코드를 안드로이드 환경에 맞도록 수정한 소스코드가 공개되어 있다. 다만, 이것만으로는 SPICE 프로토콜이 동작하지 않는다. 따라서 이 소스코드를 기반으로 SPICE 프로토콜이 동작하도록 포팅을 진행한다. 구글코드 사이트에 소개되어 있는 링크를 참조하여 진행하게 되는 SPICE 프로토콜의 포팅 과정을 간단하게 요약하면 다음과 같다[6].

- Android용 gcc 컴파일러인 agcc를 생성
- 각 모듈의 configure 이전에 ld와 ranlib를 arm-eabi-ld, arm-eabi-ranlib로 설정
- iconv, gettext, glib, pixman, jpeg, openssl 모듈을 차례대로 포팅
- androidSpice 소스코드를 ndk-build 명령으로 빌드함[7]

SPICE 클라이언트는 동작할 때 iconv, gettext, glib, pixman, openssl 모듈들을 요구한다. 위의 순서대로 포팅 작업을 진행하면 최종적으로 libspicec.so라는 ndk라이브러리 파일이 생성된다. 이 라이브러리 파일을 통해 안드로이드 SPICE 프로토콜을 실행 할 수 있다.

위의 순서대로 진행하는 중에 어떠한 문제점이 발생하여 SPICE 포팅에 지장이 생길 수 있다. 본 논문은 Ubuntu 11.10 32-bit 환경에서 발생하는 문제점과 이에 대한 처리방법 몇 가지를 제시한다.

- gettext 모듈을 포팅할 때 기존에 사용하던 함수를 제대로 호출하지 못하여 설치가 멈출 수 있다. gettext 모듈은 glib 모듈을 설치하기 위해 필요한 모듈로써 libintl.so 파일을 glib 설치 전에 제공하는 것이 목적이므로 아무 동작도 하지 않는 빈 함수를 만들어서 포팅 작업을 강제 진행시킨다.
- 각 모듈의 포팅이 끝나고 ndk-build로 공유 라이브러리를 생성할 때 모듈의 정적 라이브러리가 필요하므로 configure로 설치를 진행할 때 –static-enable 옵션을 넣어 .a 파일을 얻어야 한다.
- 테스트 전에 android-worker.c 파일에서 SPICE 클라이언트의 소켓 위치를 SPICE를 적용할 java 패키지명에 맞추어 수정한다. 예를 들어, com.keqisoft.android.spice 패키지에서 SPICE 프로토콜을 동작시키려면 소켓은'/data/data/com.keqisoft.android.spice/spice-input.socket'로 설정한다.

ndk-build 명령이 정상적으로 수행되면 libspicec.so파일이 생성된다. 안드로이드 어플리케이션에서 포팅된 SPICE 프로토콜을 사용하려면 ndk예제처럼 빌드 된 so 라이브러리 파일을 jni형식으로 읽어 들인다. java에서 jni형식으로 함수를 호출할 때 커맨드 형식으로 IP주소, Port번호, 접속암호를 인자로 전달한다. 예를 들면, "-h 111.111.111.111 -p 5930 -w password"와 같은 형태이다. jni를 통해 SPICE 클라이언트를 동작시키면 안드로이드 어플리케이션은 가상머신 서버로 원격 접속을 하게 된다. (그림 4)는 안드로이드 에뮬레이터를 통해서 윈도우즈 가상머신에 접속한 결과를 보여준다.

(그림 4) SPICE 프로토콜의 원격접속 화면
(반시계방향으로 90도 회전한 화면)

5. 결론

모바일 클라우드 서비스에 대한 요구가 증가하면서 이를 위한 다양한 기술들도 함께 발전하고 있다. 모바일 가상머신에 접속할 수 있는 모바일용 원격접속 프로토콜 또한 이에 포함된다. 원격접속 프로토콜은 크게 RFB 프로토콜과 RDP 프로토콜로 나눌 수 있으며 SPICE 프로토콜은 RDP 프로토콜과 같은 메시지 전달방식의 원격접속 프로토콜이다. 모바일 클라우드 서비스를 위해서는 트래픽 발생이 적어야 하며 클라이언트의 화면 출력의 부담이 적어야 한다. SPICE 프로토콜은 이러한 요구사항을 모두 만족하는 프로토콜로써 모바일 클라우드 환경에 가장 적합한 프로토콜로 판단된다. SPICE 프로토콜은 다양한 플랫폼을 지원하지만 아직 모바일환경은 지원하지 않는다. 본 논문에서는 리눅스용 SPICE 프로토콜을 안드로이드 환경으로 포팅하는 과정에서 발생하는 문제점을 발견하고 이를 해결하는 방법을 제시한다. SPICE 프로토콜이 안드로이드 환경에서도 동작한다면 모바일 가상머신에 대한 접속이 상당이 효율적으로 개선될 것이다.

Acknowledgement

본 논문은 지식경제부/산업기술평가관리원에서 지원하는 2012년도 산업원천기술개발사업(KI001810039260, 개인 및 기업 맞춤형 서비스를 위한 개방형 모바일 클라우드용 통합개발환경 및 이기종 단말-서버 간 협업 기술 개발)의 연구수행으로 인한 결과물임을 밝힙니다.

참고문헌

[1] ETRI, "차세대 컴퓨팅을 위한 가상화
[1] SPICE, http://spice-space.org, 2012. 09
[2] 장승주, "윈도 운영체제에서 원격 시스템 관리 프로그

램 기술 동향", 2010. 08
[3] Tristan Richardson, "The RFB Protocol", RealVNC Ltd, 2005
[4] MSDN, http://msdn.microsoft.com/en-us/library/aa383015(VS.85).aspx, 2012. 09
[5] MSDN, http://msdn.microsoft.com/en-us/library/cc241880(PROT.10).aspx, 2012. 09
[6] Porting snappy/libspicec.so ont Android-ARM, http://blog.csdn.net/rozenix/article/details/6277647, 2012. 09
[7] spice-client-android, http://code.google.com/p/spice-client-android/, 2012.10