

안드로이드용 클라우드 컴퓨팅 어플리케이션 개발을 위한 프레임워크

권용인, 양승준, 조영필, 백윤희
서울대학교 전기·컴퓨터 공학부

e-mail:yikwon, sjyang, ypcho@sor.snu.ac.kr, ypaek@snu.ac.kr

A Framework for Android Cloud Computing Application Development

Yongin Kwon, Seungjun Yang, Yeongpil Cho, Yunheung Paek
Dept. of Electrical Engineering, Seoul National University

요 약

스마트폰의 발전 속도는 어플리케이션의 복잡도 증가 속도에 못 미치며, 스마트폰에서 보다 고 사양의 어플리케이션을 수행하기를 원하는 사용자가 늘고 있다. 또한 스마트폰의 성능이 향상됨에 따라 배터리 소모와 발열량도 증가하여 이는 사용자에게 큰 부담이 되고 있다. 이러한 상황에서 클라우드 컴퓨팅은 스마트폰에서 고사양의 어플리케이션을 손쉽게 실행하도록 도와주며 스마트폰의 배터리 소모와 발열량도 줄여줄 수 있다. 하지만 클라우드 컴퓨팅 기능이 탑재된 어플리케이션을 개발하는 것은 개발자에게 큰 부담이기 때문에 본 연구에서는 안드로이드용 클라우드 컴퓨팅 프레임워크를 제안하여 어플리케이션 개발자들이 손쉽게 클라우드 컴퓨팅 기능이 탑재된 어플리케이션을 개발하도록 한다.

1. 서론

최근 스마트폰의 등장으로 인해 음악 감상이나 동영상 시청, 사진 촬영과 같은 기본적인 멀티미디어 기능 뿐만 아니라, 웹 브라우저나 게임, SNS 등 다양한 목적을 위한 복잡한 어플리케이션이 많이 개발되고 있다. 스마트폰의 성능이 빠른 속도로 향상되어 최근 판매중인 삼성 갤럭시 S2 HD LTE의 경우 CPU는 1.5GHz Dual Core가 장착되어 있고 RAM은 1GB이다.

이와 같이 스마트폰의 빠른 성능 향상에도 불구하고 어플리케이션의 복잡도는 훨씬 빠른 속도로 증가하여 스마트폰의 하드웨어 성능이 어플리케이션은 충분히 지원하지 못하거나, 어플리케이션 개발자가 스마트폰의 하드웨어 성능에 맞게 어플리케이션의 기능을 축소시켜야 하는 불편함이 있다. 스마트폰 하드웨어 성능에 적당한 복잡도의 어플리케이션을 개발한다 할지라도 스마트폰 하드웨어의 성능을 100% 활용 하다보면 배터리를 많이 소모하게 되어 방전이 빨리 되고 다량의 열을 방출하여 사용자에게 불편감을 줄 수 있다.

시중에 판매중인 스마트폰의 종류가 매우 다양하고, 그 성능 또한 천차만별이기 때문에 어플리케이션 개발자 입장에서 어느 기준으로 어플리케이션을 개발해야 할 지는 결정하기 힘든 문제이다.

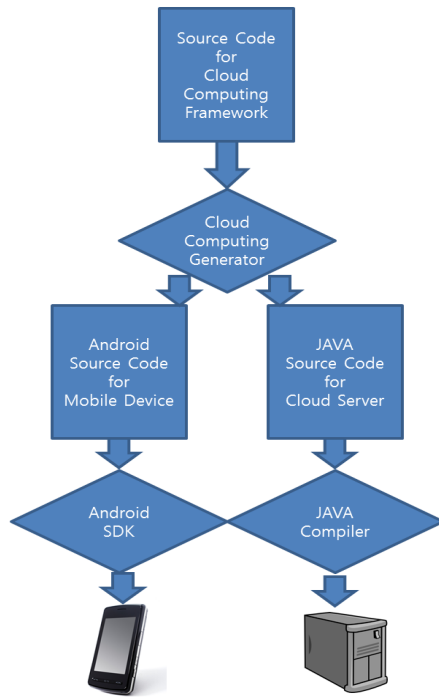
이러한 문제들을 해결하기 위한 대안 중 스마트폰과 WIFI로 연결되어있는 주변의 PC나 서버를 활용하여 어플리케이션을 클라우드 컴퓨팅하는 방법이 있다. 클라우드

컴퓨팅을 이용하면 스마트폰에서 행해져야 하는 복잡한 연산을 주변의 클라우드 서버에서 대신 수행 함으로써 높은 성능을 요하는 어플리케이션을 실행시키고, 스마트폰에서 수행할 때 보다 더 빠른 속도로 연산을 하며, 스마트폰의 배터리 소모를 줄여준다. 또한 개발단계에서 스마트폰의 성능에 큰 고려를 하지 않아도 된다.

하지만 클라우드 컴퓨팅을 가능케 하는 어플리케이션을 개발하기 위해서는 많은 노력이 필요하다. 따라서 본 연구에서는 어플리케이션 개발단계에서 쉽게 클라우드 컴퓨팅 기능을 추가할 수 있는 안드로이드용 어플리케이션 프레임워크를 제안한다.

2. 클라우드 컴퓨팅 어플리케이션 개발과정

클라우드 컴퓨팅 어플리케이션의 개발 단계는 다음 그림 1과 같다. 어플리케이션 개발자는 클라우드 컴퓨팅 프레임워크에 따라 소스코드를 작성하고 작성된 소스코드는 클라우드 컴퓨팅 제네레이터에 의해 안드로이드 기기용 소스코드와 클라우드 서버용 자바 소스코드를 각각 생성된다. 클라우드 서버용 자바 소스코드를 자바 컴파일러로 컴파일 하고 서버에서 실행시킨 후, 안드로이드 기기용 소스코드를 안드로이드 SDK를 이용하여 빌드하고 빌드된 어플리케이션을 안드로이드 기기에서 실행시키면 클라우드 컴퓨팅을 수행할 수 있게 된다.



(그림 1) 클라우드 컴퓨팅 어플리케이션 개발과정

3. 클라우드 컴퓨팅 프레임워크

클라우드 컴퓨팅 기능을 추가하여 어플리케이션을 개발하기 위해서는 클라우드 컴퓨팅 프레임워크에 맞춰서 소스코드를 작성하여야 한다.

```

public class CloudComputingTestActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        int a = 1;
        int b = 2;
        int c = foo(a,b);
        Toast.makeText(CloudComputingTestActivity.this, c,
            Toast.LENGTH_SHORT).show();
    }

    @CloudComputing
    int foo(int a, int b) {
        return a+b;
    }
}
    
```

(그림 2) 클라우드 컴퓨팅 프레임워크 예제

그림 2와 같이 클라우드 서버에서 수행되기 원하는 메소드 뒷줄에 '@CloudComputing'을 써 줌으로써 뒤의 메소드는 클라우드 컴퓨팅이 가능하도록 빌드한다는 것을 명시해줄 수 있다. 이런식으로 명시된 소스코드는 2장에서 설명했듯이 클라우드 컴퓨팅 제네레이터에 의해 안드로이드 기기용 소스코드와 클라우드서버용 자바 소스코드를 생성한다. 클라우드 컴퓨팅 제네레이터는 안드로이드 기기용 소스코드의 '@CloudComputing'이 명시된 메소드 코드를 수정하여 주변의 클라우드 서버를 찾고, 클라우드 서버에서 수행되기를 원하는 메소드명과 인자들을 넘겨준 뒤 클라우드 서버에서의 수행이 끝나면 결과값을 리턴받

아 다음 코드를 계속 수행하게 된다. 하지만 만약 주변에서 클라우드 서버를 찾지 못하면 해당 메소드를 안드로이드 기기에서 수행하도록 하여 클라우드 서버의 유무가 어플리케이션 수행에 지장을 주지 않도록 한다.

3. Discussion and Future Work

이 연구에서는 어플리케이션 개발자가 보다 더 쉽게 클라우드 컴퓨팅을 적용하여 어플리케이션을 개발할 수 있게 안드로이드용 클라우드 컴퓨팅 프레임워크를 정의하고 클라우드 컴퓨팅 제네레이터를 개발하였다. 클라우드 컴퓨팅을 원하는 메소드 앞에 '@CloudComputing'을 삽입해 줌으로써 클라우드 컴퓨팅 제네레이터가 안드로이드용 코드와, 서버용 자바 코드를 생성해준다. 하지만 지금의 버전에서는 많은 제약이 따르고, 성능도 보장을 하지 못한다. 따라서 앞으로의 연구에는 다음의 사항들이 필요하다.

- 클라우드 컴퓨팅이 가능한 코드 증가 : 현재 버전의 클라우드 컴퓨팅 제네레이터는 메소드 인자로 전달되는 변수 외에는 어떠한 힙 데이터나 스토리지 데이터를 사용할 수 없다. 이러한 코드가 삽입되어있는 경우에는 클라우드 컴퓨팅 기능이 제거된 안드로이드 코드만을 생성한다. 따라서 다양한 종류의 코드가 클라우드 컴퓨팅이 가능하도록 개선이 필요하며, 이에 따라 프로그램 분석 툴의 개발도 필요하다.
- 클라우드 컴퓨팅 메소드 프로파일링 : 해당 메소드를 프로파일링 함으로써 해당 메소드의 수행시간 등을 예상할 수 있고, 이에 따라 해당 메소드를 클라우드 컴퓨팅하는 것이 유리한지, 그렇지 않은지를 판단할 수 있다. 따라서 어플리케이션 성능이 오히려 떨어지게 되는 결과를 예방할 수 있다.

Acknowledgement

본 연구는 교육과학기술부/한국과학재단 우수연구센터 육성사업(과제번호 2012-0000470), 2011 년도 정부(교육과학기술부)의 재원으로 한국과학재단의 국가지정연구사업(No.2011-0018609) 및 IDEC 의 지원을 받아 수행되었습니다.

참고문헌

- [1] 송형주 외 4명 “인사이드 안드로이드” 위키북스
- [2] “http://developer.android.com” 안드로이드 개발자
- [3] I. Giurgiu, “Calling the cloud : Enabling mobile phones as interfaces to cloud applications” I Middleware, 2009
- [4] 민욱기 외 2명 “클라우드 컴퓨팅 기술 동향”, 전자통신동향분석, 2009