

# 클라우드 컴퓨팅을 이용한 모바일 공유 화이트보드 시스템의 설계 및 구현

경도준, 최병광, 이장호  
홍익대학교 컴퓨터공학과

e-mail : {segwin936, hogace423}@gmail.com, janghol@hongik.ac.kr

## Design and Implementation of Mobile Shared Whiteboard System using Cloud Computing

Do Jun Kyeong, Byeong-Gwang Choi, Jang Ho Lee  
Dept of Computer Engineering, Hongik University

### 요 약

최근 리소스의 규모를 자유롭게 변경할 수 있는 클라우드 컴퓨팅이 널리 쓰이고 있다. 본 논문에서는 클라우드 컴퓨팅을 이용하여 스마트폰이나 태블릿상에서 그림, 문자입력을 공유 할 수 있는 화이트보드 시스템의 설계 및 구현을 제시한다. 본 시스템의 사용자가 안드로이드 스마트폰 상의 화이트보드에서 사용자가 그림이나 문자를 그리게 되면 그 데이터는 AWS(Amazon Web Service) 클라우드 컴퓨팅서버를 통해 모든 클라이언트에게 브로드캐스트하며 그 후 클라이언트들은 받은 데이터를 자신의 스마트폰 디스플레이에 출력한다. 본 시스템은 웹서비스만으로 서버를 늘릴 수 있기 때문에 손쉽게 서버증축이 가능하다.

### 1. 서론

클라우드 컴퓨팅(Cloud Computing)이란 개인이 가진 단말기를 통해서 주로 입/출력 작업만 이루어지고, 정보 분석 및 처리, 저장, 관리, 유통 등의 작업은 클라우드라고 불리는 제3의 공간에서 이루어지는 컴퓨팅 시스템 형태로 최근 서버 구축에 널리 쓰이고 있다.

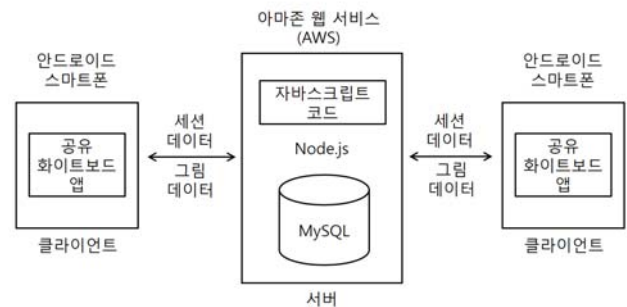
대표적인 클라우드 컴퓨팅 서비스에는 AWS(Amazon Web Service) 중에 EC2(Elastic Compute Cloud)[1]가 있다. EC2는 클라우드에서 컴퓨팅 파워의 규모를 자유자재로 변경할 수 있는 웹 서비스이다. 개발자가 보다 쉽게 클라우드 컴퓨팅 작업을 할 수 있도록 설계되어 있다.

이에 본 논문에서는 AWS에 기반한 브로드캐스팅(Broadcasting) 서버를 이용하여 안드로이드 스마트폰 상에서 사용자들간에 그림이나 문자입력을 공유할 수 있는 화이트보드 시스템을 개발하였다.

유사 시스템인 LiveBoard[5]도 안드로이드 스마트폰 사용자들이 실시간으로 공동으로 그림 작업을 할 수 있는 공유 화이트보드 기능을 제공한다.

### 2. 시스템 설계

본 논문에 제시된 시스템의 구조는 그림 1에서 보이는 바와 같이 클라이언트-서버(Client-Server)구조로 이루어져 있다.



(그림 1) 시스템 설계

본 시스템은 모델-뷰-컨트롤러(Model View Controller) 패턴을 응용하여 구성되었으며 클라이언트로부터 컨트롤러인 서버로 데이터가 전송된다. 이 때 클라이언트로부터 서버로 전송되는 데이터는 두 가지 타입으로 나뉜다. 하나는 사용자의 입장과 퇴장을 나타는 세션(Session) 데이터이며 다른 하나는 사용자가 공용 화이트보드에 그리는 그리기 데이터이다. 컨트롤러는 서버의 데이터베이스인 모델에 해당 정보를 전달하여 데이터를 관리하도록 한다. 뷰인 클라이언트에 세션 또는 그리기데이터가 있는 경우 컨트롤러가 모델로부터 정보를 받아 뷰에 전송하고 뷰는 이를 화면에 출력한다.

룸(room)을 생성하는 클라이언트를 Master라 하며, Master가 아닌 클라이언트들은 Master의 룸에 입장하여 Master의 화이트보드를 공유한다. 클라이언트들 모두 화이

트보드에 텍스트를 추가하거나 그림을 그릴 수 있다. 화이트보드에 그려진 그림들을 클라우드 서버의 데이터베이스에 저장시켜 놓고 있다가 늦게 입장한 클라이언트에게 데이터를 전송하여 클라이언트의 화이트보드에 출력한다.

### 3. 시스템 구현

#### 3.1 시스템 개발 환경

본 논문에 제시된 시스템의 개발환경은 그림 2에서 보이는 바와 같다.

개발 환경	Client	Server
OS	Android	Amazon Linux (AWS EC2)
개발언어	Java	Javascript, sql
Tool	Android studio	Sublime text2, Xshell, Mysql, Postman, HeidiSQL
기술	loopj android-async-http, socket.io-client	Node.js, Socket.io, async, express, AWS

(그림 2) 시스템 개발환경

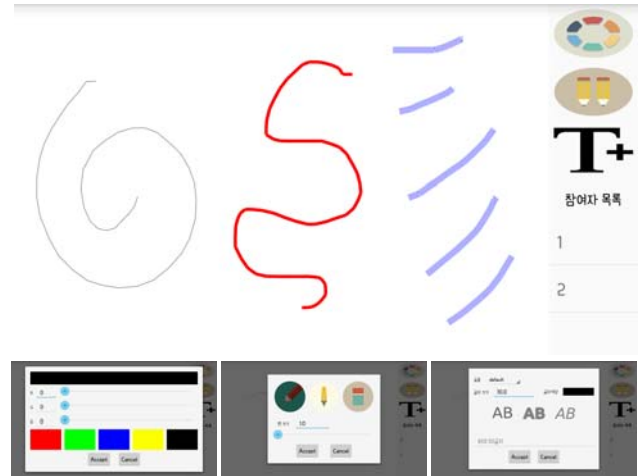
Client 운영체제는 Android로 개발 단말은 팬택 베가 시크릿 노트(A890S, 4.4.2, Kitkat), 팬택 베가 R3(A850k, 4.4.2, Kitkat), 삼성 갤럭시 노트3(SM-N900, 5.1.1 Lollipop)이며 개발 툴은 Android Studio[2]를 이용하였다. 서버 운영체제는 클라우드 컴퓨팅 서비스 AWS EC2에서 제공하는 Amazon Linux를 사용하였다. 또한 서버는 Non-blocking I/O와 단일 스레드 이벤트 루프를 통한 높은 처리 성능을 가지고 있는 Node.js[3]로 개발하였다. Node.js는 javascript로 개발하였으며 서버의 데이터베이스는 MySQL을 이용하여 개발하였다.

#### 3.2 클라이언트 구현 및 사용 시나리오

그림 3은 스마트폰 클라이언트 유저 인터페이스를 보여 준다. 클라이언트 유저 인터페이스의 대부분은 캔버스로 이루어져 있으며 오른쪽에는 색상 선택, 펜 종류 및 굵기 선택, 텍스트 입력 버튼과 방에 참여한 참여자 목록이 표시되도록 배치했다. 그림 3의 하단 스냅샷은 위에 설명한대로 색상선택, 펜 종류 및 굵기, 텍스트 입력 버튼을 누른 상태의 스냅샷이다.

로그인 화면에서 서버와 연결이 이루어지고 개설된 룸 목록을 받아 화면에 출력한다. 룸에 입장하면 그 룸의 Master 및 참여한 클라이언트들과 연결이 된다. 이 때 세션데이터와 기존에 서버의 데이터베이스에 저장되어 있던 그리기데이터를 받아 화면에 출력한다. 화이트보드에 터치 입력이 들어오는 순간 클라이언트에서 서버로 데이터가 전송되고 서버에서는 이 정보를 데이터베이스에 저장한 후

룸에 속한 Master 및 클라이언트들에게 전송한다.



(그림 3) 스마트폰 클라이언트 유저인터페이스

Master가 룸을 나갈 경우 해당 룸이 소멸되며 클라이언트 전원이 방에서 나가지게 된다.

#### 3.3 서버 구현

서버는 데이터베이스를 접근하는 부분과 Client로부터 데이터를 수신하거나 전송하는 부분을 분리하여 모듈화 한다. 모델-뷰-컨트롤러패턴을 적용하여 데이터베이스를 다루는 모델과 기능을 다루는 컨트롤러를 서버에서 구현하고 뷰는 클라이언트에서 구현 하였다.

각 화이트보드 별 룸은 Node.js의 socket.io[4] 내부에 있는 룸(Room)기능을 이용하여 구현하였다. 그리기 정보를 받아서 같은 룸 안에 있는 클라이언트들에게 브로드캐스트(Broadcast)하도록 구현하였다.

Node.js는 모듈기반의 짧은 개발시간과 단일 스레드이지만 Non-blocking I/O와 이벤트 루프를 통한 높은 처리능력을 제공하므로 효율적인 서버구축이 가능하다.

#### 4. 결론 및 향후 연구과제

본 논문에서는 클라우드 컴퓨팅 서비스인 AWS-EC2서버에 Node.js 플랫폼을 이용한 공유 화이트보드를 구현하였다. 본 시스템은 물리적인 서버를 마련하는 것보다 확장에 필요한 시간과 자원을 절약할 수 있다.

이 연구를 위해 개발한 공유 화이트보드 시스템은 음성대화기능, 파일공유기능을 추가하여 강의록을 공유하고, 음성대화 형식으로 강의를 진행할 수 있도록 연구가 진행될 것이다.

#### 참고문헌

[1] AWS EC2, <https://aws.amazon.com/ko/ec2/>  
 [2] Android Studio, <https://developer.android.com/>  
 [3] Node.js, <https://nodejs.org/ko/>  
 [4] socket.io, <http://socket.io/docs/rooms-and-namespaces/>  
 [5] LiveBoard, <http://liveboardapp.net/>