MacSeisApp:
An Early Earthquake Warning System

Shayan Mehrazarin
October 28, 2015
California State University, Dominguez Hills

Outline

- Introduction
- Background on Apple’s Sudden Motion Sensor (SMS)
- Background on Apple’s Push Notification System (APNS)
- An Inside Look at MacSeisApp
- The Role of the Sudden Motion Sensor in MacSeisApp
- Designing the Seismograph Interface
- Using Push Notifications in MacSeisApp
- Live Demonstration
- Future Work
- Questions

Introduction

- MacSeisApp is inspired by an existing project called *SeisMac* where it displays a live seismograph based on data collected by the MacBook's Sudden Motion Sensor (SMS)
  - Based on open-source library called *smslib*, which is written in the C programming language
  - *smslib* library is also used in this project to extract data collected from SMS
- The goal of MacSeisApp is to emulate a seismograph, as in SeisMac, but also add a notification system when it thinks an earthquake is occurring
  - This way, users in nearby cities are alerted seconds before seismic waves reach them
  - This is possible, as an internet connection travels faster than seismic waves

Background on Apple’s Sudden Motion Sensor (SMS)

- Apple introduced the Sudden Motion Sensor in their MacBook line of notebooks starting in 2006
- The main purpose of the SMS is to protect the MacBook’s spinning hard disk drive prior to hitting a hard surface when falling down
  - HDD receives signal to stop spinning when violent motion is detected by sensor
- The Sudden Motion Sensor measures movement along the x, y, and z axes
- Since Solid State Drives (SSDs) do not have any spinning or moving components, newer MacBook models sold within the last couple of years are not equipped with SSDs, and consequently does not include the SMS

Background on Apple's Push Notification Service (APNS)

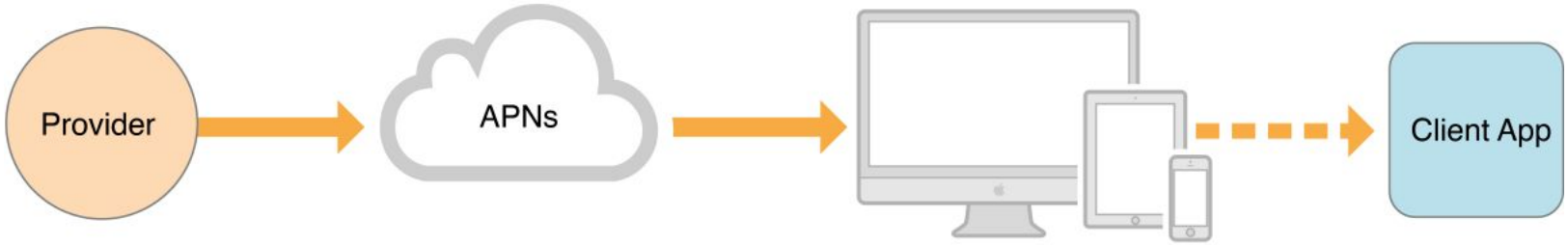
- Push Notifications were first introduced in iOS 3.0 in 2009 and Mac OS X Lion v10.7 in 2011
  - Meant to provide an alternative to existing *local* notification functionality
  - Commonly referred to as *remote* notification in technical/developer documentation
- Gives developers the capability to send notifications to users even when the application is not running
- It is used to notify users in various scenarios:
  - New or unread messages
  - Weapon unlocks or invitations in games
  - Severe weather alerts (i.e. tornado warnings, flash flood warnings)
  - Friend requests in social media

Background on Apple’s Push Notification Service (APNS)

- The basic idea behind APNS is:
  - Developer sets up a server that will run a script
  - Script communicates with Apple’s dedicated Push Notification Server using a secure connection
    - Used to be SSL, now TLS is used instead due to POODLE attack on SSL 3.0 in 2014
  - Apple’s Push Notification server verifies authenticity of developer’s server’s security certificates
  - Apple’s server then *pushes* the notification to the user’s device
- However, developers are still permitted to use SSL certificates while in development, or *sandbox* mode

Background on Apple’s Push Notification Service (APNS)

- The following diagram illustrates the general idea of APNS:



(credit: Apple Developer Program documentation)

An Inside Look at MacSeisApp

- MacSeisApp is developed using a combination of Objective-C and C code using the Xcode Integrated Development Environment (IDE)
  - This is possible, since Objective-C is based off of the C programming language
  - Objective-C vs. C code is easy to tell apart
  - In Objective-C, every project has a class called the *AppDelegate*
    - handles tasks that need to be done “behind-the-scenes”/ “under-the-hood”

An Inside Look at MacSeisApp

- This is what the AppDelegate file will typically have:

```
- (void)applicationDidFinishLaunching:(NSNotification *)aNotification {
    // Insert code here to initialize your application
    smsStartup(nil, nil);           // Used to initialize access to Sudden Motion Sensor (SMS)
    smsLoadCalibration();           // Used to load any stored calibration values
}

- (void)applicationWillTerminate:(NSNotification *)aNotification {
    // Insert code here to tear down your application
    smsShutdown();          // Shut down SMSLib once application is killed
}
```

An Inside Look at MacSeisApp

- This is some code that demonstrates the mixing of Objective-C and C code :

```
-(void) drawRect: (NSRect) bounds
{

    if(timerSet == NO)
    {
        [ NSTimer scheduledTimerWithTimeInterval:0.1f target:self selector:@selector(render:) userInfo:nil
            repeats:YES ];
        timerSet = YES;
    }

    glClearColor(0, 0, 0, 1.0);
    glClear(GL_COLOR_BUFFER_BIT);
    drawAxisLines();
    plotSeismicData();
    glFlush();


}
```

An Inside Look at MacSeisApp

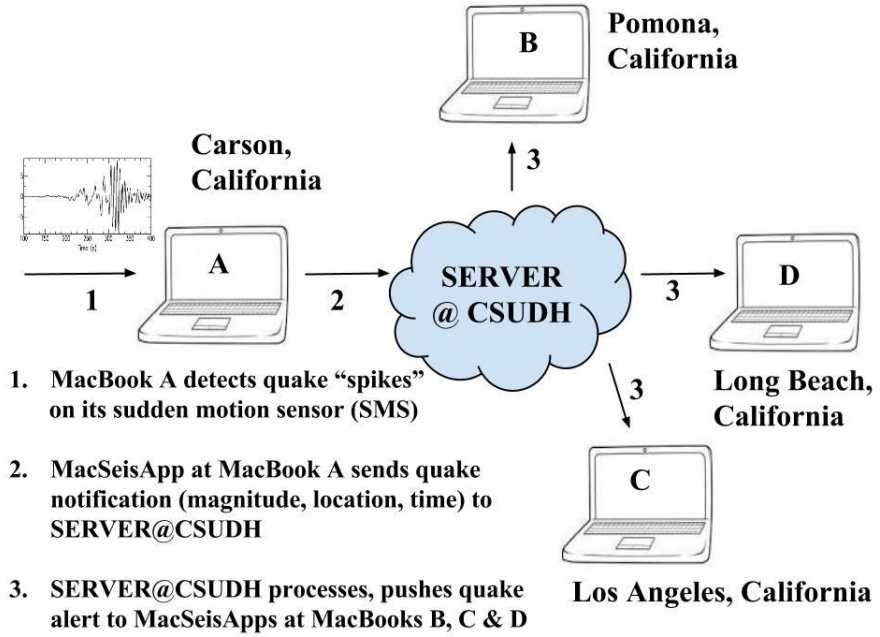
- This snippet of code shows how new data is inserted and existing data is shifted back by 1 index position via a normal C array of size 100:

```
for(int i = 0; i < 99; i++)
{                                   // Shift all values back one index space
    x_axis[i] = x_axis[i + 1];
    y_axis[i] = y_axis[i + 1];
    z_axis[i] = z_axis[i + 1];
}

x_axis[99] = x;                 // Insert new data at the end of array
y_axis[99] = y;
z_axis[99] = z;
```

An Inside Look at MacSeisApp

- The general structure and idea behind the application is shown as follows:



The Role of Sudden Motion Sensor in MacSeisApp

- The Sudden Motion Sensor is responsible for collecting motion data once every 0.1 seconds in this project
- The *smslib* library then translates the data into values of type *float* or *double*
  - Data can be accessed using provided accessor functions written in C
  - Each of the 3 axes (x, y, and z) can be used in the C and Objective-C code
- The motion data is then used by being plotted to the window on 3 separate lines to depict a live seismograph
  - Seismograph is updated each time new data is collected (once every 0.1 seconds)
  - A total of 100 plots of data is displayed at a time on each axis and is connected with lines

Designing the Seismograph Interface

- The seismograph interface was designed using the OpenGL library
  - It is an open-source library originally written in C
  - Works seamlessly together with Objective-C code

- Timer object is used to perform following functions once every 0.1 seconds:
  - Shift existing contents in array to the left by 1
  - Add new data to the end of the array (at index 99)
    - original data in index 0 is overwritten
  - Redraw the interface with the new contents of the array

Designing the Seismograph Interface

- The seismograph on SeisMac looks like this:

Designing the Seismograph Interface

- The seismograph interface previously described for MacSeisApp is shown as follows:

Using Push Notifications in MacSeisApp

- Push Notifications are used in MacSeisApp mainly to get the user's immediate attention
- If the seismograph interface detects a "spike", then a push notification will be generated and sent to the users in nearby areas
  - For example, a spike is detected on a user's MacBook Pro in Torrance
  - A push notification will be sent to users nearby in Carson, Long Beach, Los Angeles, Irvine, Riverside, etc.
  - Users in those areas will receive the notification just seconds before they feel the seismic wave
    - This is because data travels in a network faster than the speed at which seismic waves travel

Using Push Notifications in MacSeisApp

- A push notification for MacSeisApp will look something like this:

Using Push Notifications in MacSeisApp

- The push notification looks like this on a typical screen, in the top right-hand corner:

Live Demonstration

Future Work

- Since it is still in development, MacSeisApp relies on this MacBook as the “server” that communicates with Apple’s Push Notification Service
  - The next step is to deploy the script on a dedicated server on campus at CSUDH
  - This way, other devices that have MacSeisApp installed can receive push notifications
- An iOS counterpart application might be developed to also receive push notifications on iPhone, iPad devices, as well
  - Can also display latest 5 or 10 earthquake events
- Later can add an estimated earthquake rating using the Richter scale in the push notification
- Add mechanism to the algorithm that can differentiate between man-make shaking and legitimate seismic activity

Any Questions?