Attribution Framework Report

Teng Wang and Matt Bishop

September 30, 2012

System Scope

This attribution framework consists of three parts: data sensors, server and clients. Data sensors can generate data stream and send them directly to server. Server keeps monitoring communication port and stores all the data from sensors into database. As to the clients, each of them can send query commands to server and get information from server database.

In general, this framework can realize the function of gathering data distributively and support centralized data storage and query services.

System Architecture and Installation

In this project, we build our system on Emulab, a network testbed for researchers to develop, debug and evaluate their systems. On the whole, our system is made of two components:

1) Real-time data communication between sensors and server. We use Java socket programming to establish communication link and send data stream from each senor to the server.

2) Database storage and remote query. We select MySQL as our database and use JDBC API to control its data format conversion and remote query behavior.

To make it easy to use, we have integrated all the software installation and configuration into a customized OS image. The image's name is *UbuntuOS4forGENI* and its PID is *GENIHiveMind*. Any authorized Emulab user can easily use this OS image to create their own experiment with the following statement:

tb-set-node-os $nodeA UbuntuOS4forGENI

In order to build an attribution framework on Emulab, we use the same customized OS image for all the sensors, server and clients. Each machine can perform different functions depending on the different programs running on it: Data sensors have *MyClient.java* running on it; Server has *MyServer.java* running on it; Clients have *MyClient.java* running on it.

Among all the components in our system, server is different from others because of the MySQL database installed on it. All the data got from sensors are stored in table *data* , database *Framework*. Users can use basic SQL sentence to check its data:

use Framework;
*select * from data;*

After making different machines running different programs depending on their functions, the installation step is finished.

Human-Machine Interface

Now we use 4 machines (2 sensors, 1 server and 1 client) on Emulab to show a small demo of our attribute framework:

SensorA

When file *MyClient.java* is running on SensorA, user can input data and sent it to server with the format:

*AttributeName;***;AttributeValue;***;Entity;***;*

The example command is as follows(Figure 1).

```
root@nodeb:/users/Teng/www/javasocket/javafile# java MyClient
AttributeName;sys;AttributeValue;ios6;Entity;IP = 5.5.5.5
```

Figure 1: SensorA

SensorB

The data format on all the sensors are the same. User can input data from different sensors and save them on server(Figure 2).

```
root@nodec:/users/Teng/www/javasocket/javafile# java MyAnotherClient
AttributeName;os;AttributeValue;win8;Entity;IP = 7.7.7.7
```

Figure 2: SensorB

Server

When *MyServer.java* is running on server, it will keep monitoring the communication port. After server receives the data from clients, it will store it into MySQL database and show it on the screen at the same time. For example, after it gots two data packets from ClientA and ClientB, it will show its received data (Figure 3).

```
root@nodea:/users/Teng/www/javasocket/javafile# java MyServer
AttributeName;sys;AttributeValue;ios6;Entity;IP = 5.5.5.5
AttributeName;os;AttributeValue;win8;Entity;IP = 7.7.7.7
```

Figure 3: Server

Client

User can send remote query to server and get data from server database at any time. Its data format is the same with basic SQL sentence.

When *QueryData.java* is running on Client, it will first print out a sentence(Figure 4): Please input your SQL Query(use ”exit” to jump out).

```
root@noded:/users/Teng/www/javasocket/javafile# java QueryData
Please input your SQL Query (use "exit" to jump out):
```

Figure 4: Client1

Then user can start to query data from remote client(Figure 5 and 6).

```
root@noded:/users/Teng/www/javasocket/javafile# java QueryData
Please input your SQL Query (use "exit" to jump out):
select Entity from data where AttributeValue = 'ios6';
Your query sentence is: select Entity from data where AttributeValue = 'ios6';
Database Connected!

Result Row No.1
AttributeName: IP = 5.5.5.5
```

Figure 5: Client2

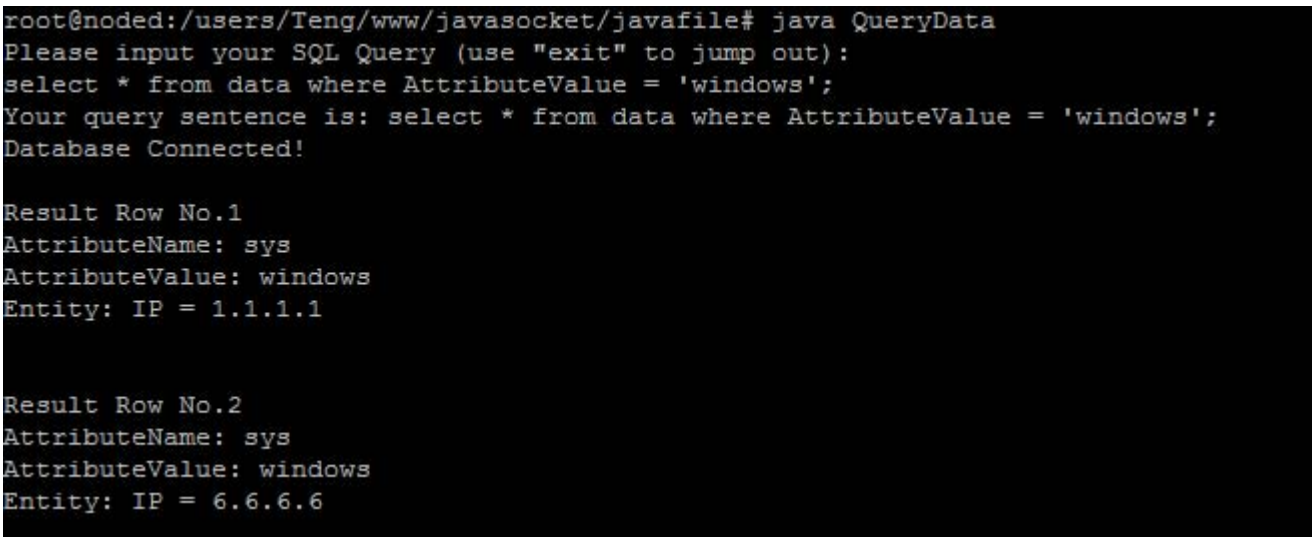

Figure 6: Client3