Alex Mahlmeister
Dr. Arias
Software Development I
6 May 2017

Final Write-Up: GeoLocation Finder Powered by iTraceIP

Abstract:

The final project, GeoLocation Finder Powered by iTraceIP, is a networking geared program that enables a user to visualize the ping process and to check packet statics. The user interface (U.I.) of the program was designed to be simple so that anyone, regardless of technical background, can use it. All that is required of the user is a web address and the program does the rest. Some core functions of the program are that it checks the validity of the address, pings said address, and displays statistical information such as packet latency and packet loss. In addition to these functions, the program will produce a rendering of the users IP location and the location of the ping server on a Google API Map.

Introduction:

The motivation to create GeoLocation Finder Powered by iTraceIP was inspired by an internship I did at a technology company working with routers, VPNs, and IPs in different company locations. One task I had to do was categorize a block of IPs and figure out what IP belongs to what office location and if they were being utilized. I thought it would be an intuitive idea if I created a program that not only pings the address, but also maps out where the server is located. The purpose of this program is to give the user a dual view of the ping process by giving them the ping statics as well as a visualization of the location of the server the packet request is ending up.

GeoLocation Finder Powered by iTraceIP is a predominantly Java-based program that uses a Java backend and a JSP front end. The JSP front end is written in HTML, CSS, and JavaScript to have a simple U.I. and to add an aesthetic appeal. The JSP page makes the program easier to use in regards that the user is not working with the Java code directly, instead, the user is working on a web page to get the information from the console.

Detailed Systems Description:

A user would use GeoLocation Finder Powered by iTraceIP when they want a dynamic experience when pinging a web address. The program itself is designed to enact a ping command from a user's website input and then display the locations of the user and the web address on a map.

The GeoLocation Finder is a user-friendly program and does not require much technical knowledge to operate. The program will prompt the user to input a website on a JSP page that is easy to navigate and follow. The JSP page allows for the user to interact with a familiar looking page than working in the terminal with than Java code directly. Inside the text box prompt, the user can enter a web address that they wish to process and the page gives an example address to show the user the proper input protocol. Once the user enters the address, they click the "Find GeoLocation" button at the bottom of the page that initializes the Java backend.

The program will then proceed to display in the console the users' location as well as the web address' location including city, state, zip code, latitude, and longitude. After that information is found, then the program will execute the ping request to produce four packet transactions and display the statics associated with the request. From there, another JSP page would be rendered in place of the initial one, this page showing a Google API map with markers showing the location of the user as well as the location of the website server connected by a polyline. After the request is executed, the user can scroll throw the console and evaluate the collected data and use the Google Maps by interacting with the map by scrolling around and zooming in or out.

Requirements:

The GeoLocation Finder Powered by the iTraceIP program is designed to make the ping command a more versatile and dynamic tool that can be utilized in a numerous amount of ways. Its application in the corporate setting is to have a more holistic image of a ping command and to offer more information about a web address or IP than what a traditional ping command would provide. The program states and displays then web address servers' location in retrospect to the users' location.

The program in the academic setting could serve as a learning tool, teaching students or whomever what a ping is and what it specifically does. It enables a person to turn plain terminal data into visual data so they can see where their command prompt is coming from and going to and the latency of the request. The program can be used in a various amount of ways and is not bound to one particular setting.

Literature Survey:

There are other ways to achieve the same results that are outputted by the GeoLocation Finder Powered by iTraceIP, but it is more tedious and cumbersome to attain. The user can use the traditional command line and get the IP and packet information there, but then they would need to go to a geo IP website to look up the location of the IP. This methodology is inefficient when compared to that of the GeoLocation Finder because the program gathers and displays the information on a single page rather than having the user look up additional IP information.

In addition to this, GeoLocation Finder can be used without having any knowledge of working a command line prompt. It enables people who have no technical skills to execute a ping command and gather details about a particular request.

User Manual:

To run GeoLocation Finder Powered by iTraceIP, the user needs to run an Apache server (I use Apache Tomcat/7.0.37) and the GeoLiteCity-2 database.

When the user initially starts up the GeoLocationFinder.jsp on the Apache server, they are greeted with a JSP page that is designed so that the user knows exactly what to do. On the JSP page the name of the program is on the top of the page and then there is a text box below the name stating for the user to input a web address following a specified protocol. After that, there is a giant green button that says "Find GeoLocation," and once the user clicks it, it will start the Java backend.

From the JSP frontend, the Java backend is initialized when the JSP initiated the 'get' method from the GeoLocationFinderServlet. From there, if the web address does not meet the protocol of the program, it will throw a 'null pointer' exception. If the web address is validated, then the method reads the current user's IP address and gets the web address from the request parameter. It then passes the user IP and web address to the getLocation method to retrieve the GeoLocation data from the imported database. After retrieving the location data, it adds the response from the method to a HashMap and forwards to the caller method.

Then the web address is run through the system command method to retrieve the ping status of four packets, which is run on the console of the computer. The results of the ping are then stored in a string builder and displayed in the console as well as the locations of the user and the website address. Then another JSP page uses a Java scriptlet to get the attribute values from the servlet request so the values could be implemented in the JavaScript code to renders a Google API map according to the latitude and longitude of both the users' location and the server location.

Conclusion:

Overall, GeoLocation Finder Powered by iTraceIP is designed to be a multifaceted tool that can be used in a copious amount of places. It can serve as an efficient way to not only ping the desired address, but also to learn more about said address and visually see the processes take place. The program accomplishes its goal of providing the user with a dynamic experience and enables a wider range of people to execute command prompts without knowledge of how to use the console.

References:

https://gist.github.com/madan712/4509039 (Ping Status)

https://www.mkyong.com/java/java-find-location-using-ip-address/ (GeoLocation Finder)

https://developers.google.com/maps/documentation/javascript/ (Google Maps API)

https://www.tutorialspoint.com/servlets/servlets_overview.htm (Servlet Information)