

Geoplot - The World Map Display

The world map is created by using the Geoplot applet from CAIDA. Each monitoring host has a directory \$REPORTSDIR/\$aliasName/geoplot where the html, jar, and configuration files are stored. For example: for the monitoring host IEPM-BW, the directory is /nfs/slac/g/net/iepm-bw/iepm-bw.slac.stanford.edu/geoplot. The \$REPORTSDIR is "/nfs/slac/g/net/iepm-bw/", \$aliasName is "iepm-bw.slac.stanford.edu", and the geoplot directory is "geoplot". The relevant files in this directory are:

- GeoPlot.Jar - the jar file
- geoplot.html - The HTML file
- World100.gif - the background world map file
- iepmlocations.txt - the locations of the nodes - created by \$iepmSrcDir/fetch-lat-long
- update-geoplot.txt - the file specified in the field of the applet where the plotting instructions are stored. This file is updated by the \$iepmSrcDir/map-updated which is running as a daemon on the monitoring host.

To modify the look (add more info, etc) to the main plot page, modify the geoplot.html file. GeoPlot is a light-weight java applet which allows users to create a geographical image of a data set. The applet provides the user with many options to represent the data set. Basically, GeoPlot plots a set of nodes and a set of lines that connect these nodes on an image specified by the user. To deploy Geoplot on any web-server, we need three files.

GeoPlot.jar

HTML to display the GeoPlot applet. The image file to display on the applet.

Geoplot.jar

The Geoplot.jar file consists of six java code files. Geoplot.java Key.java Line.java Node.java NodLin.java StatusBar.java

Geoplot.java file get nodes and lines data to display from a URL specified in applet parameters in the HTML page. In our case when GeoPlot.java was trying to get information from that specified URL it sometime comes up with blank-lines. This problem was crashing the whole application. So we decided to modify the GeoPlot.java code.

In Geoplot.java code there is a method parseData. This method is used to parse the data creating a node and line object for node/line definition. If the data is specified as a URL, then the functions reads the file and then does the parsing. `str = tokenizer.nextToken(); str = str.trim(); ch = Character.toLowerCase(str.charAt(0));`

The above mentioned code in the parseData method is responsible to get one line of that data file and get the first character to check what its value is. Line 1 gets the next token. Line 2 trims the token to remove spaces. Line 3 gets the character at location 0 and covert it to lowercase. Now it is quite possible that the token which we are trying to get in line 1 has null value. So we modified the code to following `str = tokenizer.nextToken(); str = str.trim(); if (str.length() == 0) continue; ch = Character.toLowerCase(str.charAt(0));`

The only addition which we made to the code was to add Line 4. Line 4 actually say that if the length of the line is 0 don't process that line and continue with the next one.

HTML to Display Applet

Place the following applet tag in your html file. GeoPlot.class is the main class file that should be placed in your applet tag. The basic outline is shown below:

```
< applet code="GeoPlot.class" archive="GeoPlot.jar" width=??? height=???>
< param name="World100.gif" value="World100.gif">
:
:
:
:
< /applet>
```

The width and height attributes of the applet tag refer to the width and height of the applet in the browser window. All the param tags follow the opening applet tag.

Image File

Image file will be displayed as a background on the applet. This image will contain all the nodes and lines on it.

Adding nodes

Nodes are added by editing the file iepmlocations.txt. This contains one line for each node.

Contributed by Fawad Nazir.