



## Using the Google Maps API for Flow Visualization

*Where on Earth is my Data?*

**Sid Faber**  
**Network Situational Awareness Group**  
[sfaber@cert.org](mailto:sfaber@cert.org)



Software Engineering Institute | Carnegie Mellon

© 2007 Carnegie Mellon University



# Agenda

---

- Step 1: Extracting Flow Data
- Step 2: Geolocation
- Step 3: Convert to XML
- Aside: The Google Maps API
- Step 4: The HTML Page




# Data Used for Demo

---



## SC06 Data Set

- November 14, 2006
- Goal is to look at who talked to whom





**Step 1:**  
*Extracting Flow Data*

  Software Engineering Institute | Carnegie Mellon

© 2007 Carnegie Mellon University 4

## Extracting Flow Data

---

What story do you want to tell with geolocation?

- Traffic source or destination
  - Data record = one value per address
- Relations between addresses
  - Data record = one value per source, destination address pair



# Extracting Flow Data: SiLK Example

## Traffic destination

```
$ rfilter
  --start=2006/11/14
  --proto=0-255
  --class=all --pass=stdout
  | runiq
  --fields=dip --bytes > dst.txt
140.221.159.103 12568504471655
172.30.5.11 11381325217792
172.30.6.11 7397483692032
```



## Step 1: Summary

---

### Extract Flow Data

- Start with raw flow data
- End with summarized flow data (2 columns)
  - Destination IP, value
  - Space delimited

- For Example:

```
140. 221. 159. 103  12568504471655  
172. 30. 5. 11    11381325217792  
172. 30. 6. 11    7397483692032
```





## Step 2:

*Geolocating IP Addresses*

**CERT**



Software Engineering Institute | Carnegie Mellon

© 2007 Carnegie Mellon University

8



# Geolocating by Country

Map IP to Country: IPelligence, <http://www.ipelligence.com>

"0000000000", "0033554431", "US", "UNITED STATES", "NA" . . .  
"0033554432", "0050331647", "DE", "GERMANY", "EU", "EUROPE"

Map Country to Lat/Long: MaxMind, [http://www.maxmind.com/app/country\\_latlon](http://www.maxmind.com/app/country_latlon) *Numeric IP*

US, 38.0000, -97.0000  
DE, 51.0000, 9.0000

Combine IP-to-Lat/Long Mapping

0000000000 0033554431 US 38.0000 -97.0000  
0033554432 0050331647 DE 51.0000 9.0000  
0050331648 0067108863 HK 22.2500 114.1667



# Geolocating by Addresses

## DNS LOC

```
$ host -t LOC cmu.edu  
    cmu.edu LOC 40 26 39.000 N 79 56 36.200 W 283.00m ...
```

## Caida Netgeo

```
$ wget http://netgeo.caida.org/perl/netgeo.cgi \  
?target=128.2.10.162
```

```
...  
TARGET:      128.2.10.162<br>  
NAME:        CMU-NET<br>  
NUMBER:      128.2.0.0 - 128.2.255.255<br>  
LAT:         40.44<br>  
LONG:        -79.95<br>  
...
```

Hostip.info, <http://www.hostip.info/dl/index.html>



## Sample Commercial Data: Quova

1	start_ip_int	50331648	67272896
2	end_ip_int	50378239	67272959
3	cidr	24	26
4	continent	north america	north america
5	country	united states	united states
6	country_iso2	us	us
7	country_cf	80	97
8	region	northeast	northeast
9	state	connecticut	massachusetts
10	state_cf	10	87
11	city	fairfield	woburn
12	city_cf	10	77
13	postal_code	06825	01888
14	phone_number_prefix	203	781
15	timezone	-5	-5
16	latitude	41.1753	42.4867
17	longitude	-73.2812	-71.1543
...		...	...

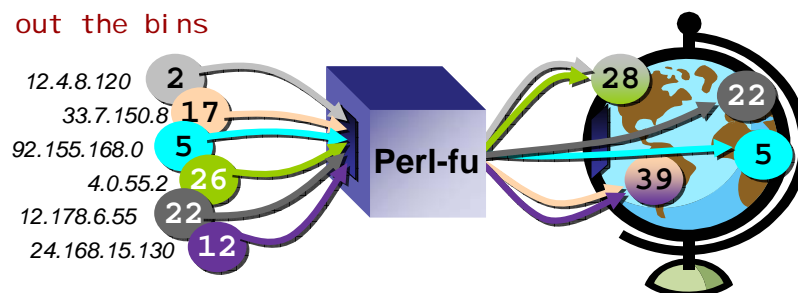
*Numeric IP*



## Add location to data and regroup

Perl-fu pseudocode:

```
Read location data into a lookup table
For each line of data {
  Extract IP and [value]
  Find lat, long coordinates for IP
  Create a bin for the coordinates and add [value]
}
Print out the bins
```



## Geolocating with SiLK pmaps

Prefix maps associate a value with an IP address prefix

- Text based pmap:

```
#Start-IP End-IP CC Lat Long pmap value
0033554432 0050331647 DE 51.0000 9.0000
0050331648 0067108863 HK 22.2500 114.1667
```



## Building the Geolocation pmap

---

Some perl-fu:

```
read countrylatlng.txt into a hash
foreach line in the iplicence data set {
  look up the countrylatlng.txt line for
  the code
  print out the ip range, country code and
  coordinates
}
```

- See *make-geo-cc-pmap.pl* in the sample code



## Using the Geolocation pmap

Use the pmap with rwuniq:

```
$ rfilter \  
  --start=2006/11/14 \  
  --proto=0-255 \  
  --class=all --pass=stdout \  
 | rwuniq \  
  --pmap-file=geo-cc.pmap \  
  --fields=dval --bytes --delimited=" " --no-titles \  
> geo-dst.txt  
  
US 38.0000 -97.0000 102372319236580  
JP 36.0000 138.0000 9965004709495  
CA 60.0000 -95.0000 569989239278
```



## Step 2: Summary

---

### Geolocate Flow Data

- Start with summarized flow data
- End with location data (4 columns)
  - Destination label, latitude, longitude, value
  - Space delimited
  - SiLK pmaps combine steps 1 and 2
- For example:


```
US 38.0000 -97.0000 102372319236580
JP 36.0000 138.0000 9965004709495
CA 60.0000 -95.0000 569989239278
```







**Step 3:**  
*Convert to XML*

  Software Engineering Institute | Carnegie Mellon

© 2007 Carnegie Mellon University 17

# XML Data

---

## Convert to XML

- The GoogleMaps routine we'll be using takes XML input
- We define the schema
- We'll process Step 2 data with a simple awk command

```
$ cat geo-dst.txt | \
awk ' BEGIN {print "<markers>"} \
{ printf "<marker lbl=\"%s\" lat=\"%s\" lng=\"%s\" \
val=\"%s\"/> \n", $1, $2, $3, $4} \
END { print "</markers>"} ' \
> geo-dst.xml
```



## Step 3: Summary


---

### Convert to XML


- Start with labels, coordinates and values
- End with XML document with the same data
  
- For example:

```
<markers>
<marker lbl="CN" lat="35.0000" lng="105.0000" val="704206" />
<marker lbl="MR" lat="20.0000" lng="-12.0000" val="200" />
<marker lbl="KN" lat="17.3333" lng="-62.7500" val="646" />
</markers>
```



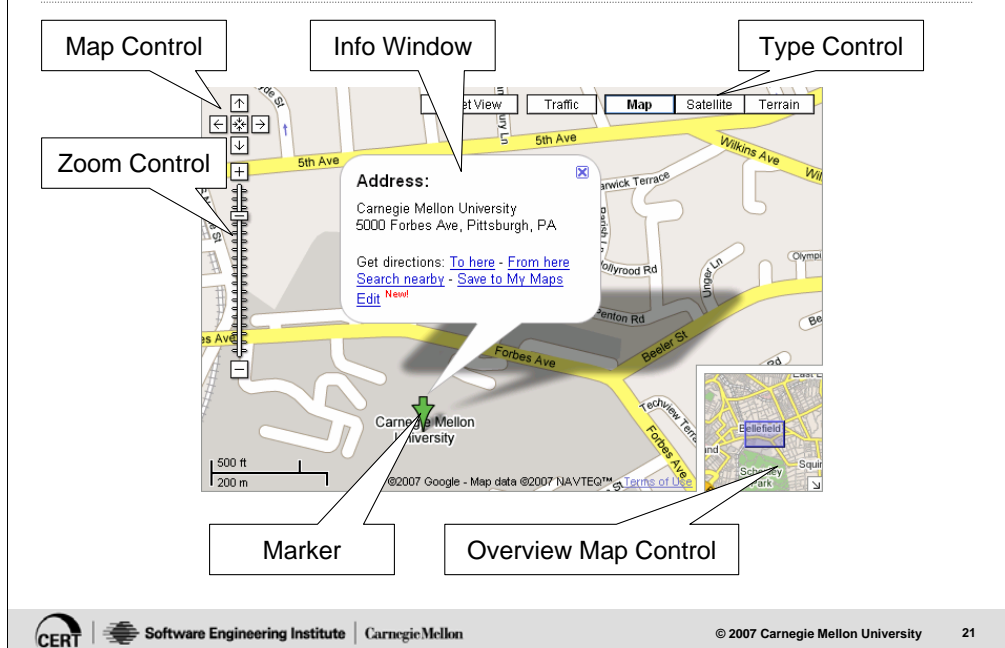


**Aside:**  
*The Google Maps API*

  Software Engineering Institute | Carnegie Mellon

© 2007 Carnegie Mellon University 20

# Google Maps Widgets



# Google Maps API Fundamentals

<http://code.google.com/apis/maps/documentation/>

- Very well documented, lots of examples
- Start simple (like this demo)
- Requires very basic javascript and HTML knowledge

## General flow:

- Include the source code
- Create the map
- Drop markers onto the map



Software Engineering Institute | Carnegie Mellon

© 2007 Carnegie Mellon University 22

## About keys and data

---

In order to include the library source, you need a key

- The key uniquely identifies your URL
- Not necessary when serving via a file:// URL

Doesn't the data get posted up to Google?

- No, Google only sees you requests for the underlying map images
- All marker placement and labeling is done local to the client with overlays





## Step 4:

*The HTML Page*

  Software Engineering Institute | Carnegie Mellon

© 2007 Carnegie Mellon University 24



## geo-dst.html (part 1)

```
<html ><head><title>IP Geolocation Example</title>
<script src="http://maps.google.com/maps?file=api&v=2&key="
    type="text/javascript"></script>

<script type="text/javascript">
// This is the file that contains the point data
var map;
var xmlFile = "geo-dst.xml";
// Called when the map is loaded. This function
// creates the map, adds controls to it, and then
// the points are laid on top of the map
function load() {
    if (GBrowserIsCompatible()) {
        map = new GMap2(document.getElementById("map"));
        map.addControl(new GLargeMapControl());
        map.addControl(new GOverviewMapControl());
        map.addControl(new GMapTypeControl());
        map.setCenter(new GLatLng(38, -97), 1);
        loadpoints();
    }
}
}
```



## geo-dst.html (part 2)

```
// http://code.google.com/apis/maps/documentation/services.html#XML_Requests
function loadpoints() {
  GDownloadUrl(xmlFile, function(data, responseCode) {
    var xml = GXml.parse(data);
    var markers = xml.documentElement.getElementsByTagName("marker");
    for (var i = 0; i < markers.length; i++) {
      var point = new GLatLng(parseFloat(markers[i].getAttribute("lat")),
        parseFloat(markers[i].getAttribute("lng")));
      descr = markers[i].getAttribute("label") + " " + markers[i].getAttribute("value");
      map.addOverlay(new GMarker(point, {title: descr, clickable: false}));
    }
  });
}
</script></head>

<body onload="load()" onunload="GUnload()"><h2>IP Geolocation Example</h2>
  <div id="map" style="width: 640px; height: 480px"></div>
</body>
</html>
```



# The Results...

## IP Geolocation Example



Software Engineering Institute | Carnegie Mellon

© 2007 Carnegie Mellon University

27

# Customizing Marker Icons

---

Two modifications needed

- Define the different icons upon initialization
- Choose the icon when points are added



## geo-dst-v2.html (part 1)

```
...
function load() {
  if (GBrowserIsCompatible()) {
    map = new GMap2(document.getElementById("map"));
    map.addControl(new GLargeMapControl());
    map.addControl(new GOverviewMapControl());
    map.addControl(new GMapTypeControl());
    map.setCenter(new GLatLng(38, -97), 1);

    //create different pins
    sredi.con.image = "green-s.png";
    sredi.con.shadow = "shadow-s.png";
    sredi.con.iconSize = new GSize(8, 13);
    sredi.con.shadowSize = new GSize(14, 13);
    sredi.con.iconAnchor = new GPoint(4, 12);
    sredi.con.infoWindowAnchor = new GPoint(5, 1);

    mredi.con.image = "red-m.png";
    mredi.con.shadow = "shadow-m.png";
    mredi.con.iconSize = new GSize(12, 20);
    ...
    loadpins();
  }
}
```



## geo-dst-v2.html (part 2)

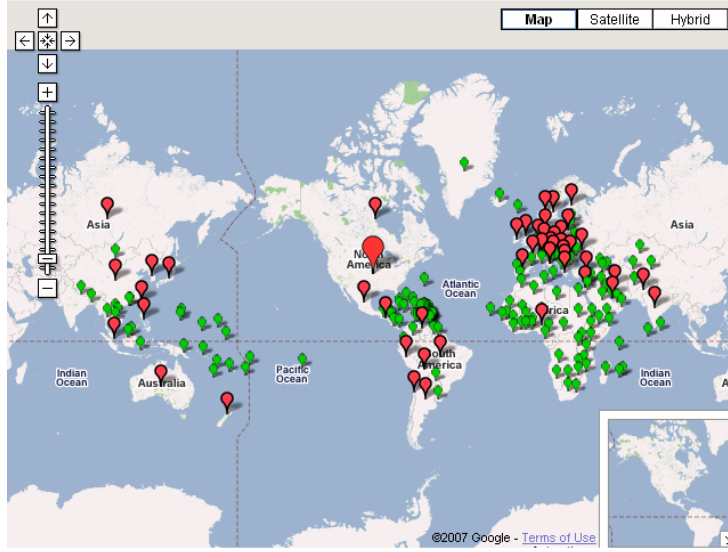
```
// http://code.google.com/apis/maps/documentation/services.html#XML_Requests
function loadpoints() {
  GDownloadUrl(xmlFile, function(data, responseCode) {
    var xml = GXml.parse(data);
    var markers = xml.documentElement.getElementsByTagName("marker");
    for (var i = 0; i < markers.length; i++) {
      var point = new GLatLng(parseFloat(markers[i].getAttribute("lat")),
                              parseFloat(markers[i].getAttribute("lng")));
      ...

      var ratio = Math.log ( parseFloat(markers[i].getAttribute("val")) /
                             minval) / Math.log (maxval / minval);
      // Plot the pin corresponding to the logarithmic ratio
      //
      if (ratio < 0.2) {
        map.addOverlay(new GMarker(pointList[i], {icon: srediCon, title: de...
      } else if (ratio < 0.9) {
        map.addOverlay(new GMarker(pointList[i], {icon: mrediCon, title: de...
      } else {
        map.addOverlay(new GMarker(pointList[i], {icon: lrediCon, title: de...
      }
    }
  }
  ...
}
```



# The Results...

## IP Geolocation Example



Software Engineering Institute | Carnegie Mellon

© 2007 Carnegie Mellon University

31

## Adding Links

---

### Need a new data set

- Create an XML file with source location, destination location and value
- Add a new function to read and plot the data file





# geo-dst-v3.html

```
function loadLinks() {  
  GDownloadUrl(xmlFile, function(data, responseCode) {  
    ...  
    var sLink = new GLatLng(parseFloat(links[i].getAttribute("slat")),  
                             parseFloat(links[i].getAttribute("slng")));  
    var eLink = new GLatLng(parseFloat(links[i].getAttribute("elat")),  
                             parseFloat(links[i].getAttribute("elng")));  
    map.addOverlay(new GPolyline([sLink, eLink],  
                                  "#000000", ratio * 5, ratio / 2, {geodesic: true}));  
    ...  
  }  
}
```

*Color*

*Opacity*

*Thickness*



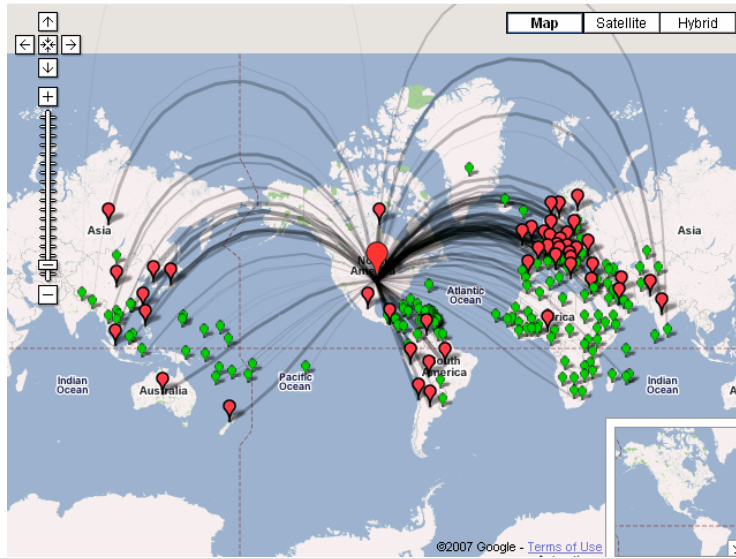
Software Engineering Institute | Carnegie Mellon

© 2007 Carnegie Mellon University

33

# The Results...

## IP Geolocation Example



Software Engineering Institute | Carnegie Mellon

© 2007 Carnegie Mellon University

34

## Where to go from here

---

Make it your own

- Generate info window popups
- Drag markers
- Add driving directions

See <http://code.google.com/apis/maps/>

Download sample code from the training server  
(128.2.243.104) in /home/sfaber/presentation



Software Engineering Institute | Carnegie Mellon

© 2007 Carnegie Mellon University

35



## Using the GoogleMaps API for Flow Visualization

*Where on earth is my data?*

**Sid Faber**  
**Network Situational Awareness Group**  
**sfaber@cert.org**

*Download sample code from the training server (128.2.243.104)  
in directory /home/sfaber/presentation*



Software Engineering Institute | Carnegie Mellon

© 2007 Carnegie Mellon University