Programming in Java
Jacobs University Bremen
Dr. Jürgen Schönwälder

Course: 320341
Date: 2015-11-29
Due: 2015-12-08

Problem Sheet #5

Problem 5.1: *storing flow data in a database* (5+5 = 10 points)

In this assignment, you are going to further extend the network flow processing program that you have written for the previous assignments. The goal of this assignment is to make data persistent using relational database tables. following extensions should be implemented:

a) In the previous assignment, you were augmenting flow records with information to which Autonomous System (AS) the IP addresses belong. To obtain this information, you were querying a REST service provided by RIPE and you maintained the responses in a non-persistent cache. Your task now is to make this cache persistent by writing responses into a relational database table. The table should have the following structure:

```
CREATE TABLE "as" (ip VARCHAR(42),
                   prefix VARCHAR(46) not NULL,
                   asn INTEGER not NULL,
                   holder VARCHAR(255))
```

Add new rows to the table whenever you have successfully looked up an IP address. Store the table in a SQLITE database 'ripe.db'. Create the table in the SQLITE database 'ripe.db' if it is not present yet.

b) Write the results produced by your program into an SQLITE database called 'flows.db'. Create a separate table for each flow filter / grouping that your program computes. For example, if you have a filter that is filtering out all flows running over TCP port 80 or 443, then the result records may go into a table named `http_flows`. If you group this traffic by AS numbers, then the results may go into a table called `http_flows_by_as`. The tables should have the following structure:

```
CREATE TABLE http_flows(time INTEGER not NULL,
                        "group" VARCHAR(255),
                        type VARCHAR(4) not NULL,
                        min INTEGER not NULL,
                        q1 INTEGER not NULL,
                        median INTEGER not NULL,
                        q3 INTEGER not NULL,
                        max INTEGER not NULL,
                        size INTEGER not NULL,
                        sum INTEGER not NULL)
```

With this setup, it is possible to plot graphs using `gnuplot` by extracting data from the database file on the fly. Below is an example `gnuplot` script.

```
set terminal pdf
set grid
set timefmt "%s"
set xdata time
set format x "%H:%M:%S"
set datafile separator "|"

set multiplot layout 2,1 title "IPv6 median flow record data (10 minute time bins)"

set yrange[0:15000]
set ylabel "octets"
plot ’< sqlite3 flows.db "select time, median from quic_flows where type = \"octs\""’ \
         using 1:2 with linespoints title "quic", \
     ’< sqlite3 flows.db "select time, median from http_flows where type = \"octs\""’ \
         using 1:2 with linespoints title "http", \
     ’< sqlite3 flows.db "select time, median from icmp6_flows where type = \"octs\""’ \
         using 1:2 with linespoints title "icmp6", \
     ’< sqlite3 flows.db "select time, median from other_flows where type = \"octs\""’ \
         using 1:2 with linespoints title "other"

set ylabel "packets"
set yrange[0:100]
plot ’< sqlite3 flows.db "select time, median from quic_flows where type = \"pkts\""’ \
         using 1:2 with linespoints title "quic", \
     ’< sqlite3 flows.db "select time, median from http_flows where type = \"pkts\""’ \
         using 1:2 with linespoints title "http", \
     ’< sqlite3 flows.db "select time, median from icmp6_flows where type = \"pkts\""’ \
         using 1:2 with linespoints title "icmp6", \
     ’< sqlite3 flows.db "select time, median from other_flows where type = \"pkts\""’ \
         using 1:2 with linespoints title "other"
```