Small Scale Data Mining
Fun and Games With Open Source

Dr E.W. Lisse

Namibian Network Information Centre (Pty) Ltd

2013-07-15
Outline

What is this Talk about?

1. Domain Name Transfers
2. Transaction Patterns
3. Registration Patterns
4. Discussion
So, What’s This?

.NA® Domain Name Transfers
One Reseller moved between Registrars repeatedly.
Background

One Reseller moved between Registrars repeatedly.

Issue

Are there Transfer Patterns?
# .NA® Domain Name Transfers

## Background
One Reseller moved between Registrars repeatedly.

## Issue
Are there Transfer Patterns?

## Tools Used
- Graphviz
- Perl
  - DBI
  - GraphViz2
- SSH Tunnel Manager
- LyX/LATEX
  - Beamer
- librsvg
Define some $variables, an @array and a %hash

```perl
my $i = 0;
my $registrar = '';
my @colors = ("aquamarine", "blue", "blueviolet", "brown",
              "violetred", "wheat", "yellow", "yellowgreen");
my %farbe;
```
Define some $variables, an @array and a %hash

```perl
my $i = 0;
my $registrar = '';
my @colors = ("aquamarine", "blue", "blueviolet", "brown",
              "violetred", "wheat", "yellow", "yellowgreen");
my %farbe;
```

Prepare GraphViz2

```perl
use GraphViz2;
my $PENWIDTH = 3;
my $driver = 'dot';
my($graph) = GraphViz2 -> new
(
    edge => {penwidth => $PENWIDTH, fontsize => 20 },
    global => {driver => $driver, directed => 1 },
    node => {penwidth => $PENWIDTH, fontsize => 11 },
);
```
Connect to the database

```perl
use DBI; my $EPP = DBI->connect("dbi:Pg:dbname=epp");
```
Connect to the database

```perl
use DBI; my $EPP = DBI->connect("dbi:Pg:dbname=epp");
```

Construct the Query

```perl
my $TRANSFERS = 5;
my $sql = q{SELECT t.owner_clid, t.transfer_to_clid, COUNT(t.*)
FROM transfer_request t
WHERE t.response LIKE '%APPROVED'
GROUP BY t.transfer_to_clid, t.owner_clid
HAVING COUNT(t.*) > $TRANSFERS
ORDER BY t.owner_clid, t.transfer_to_clid};
```
### Connect to the database

```perl
use DBI; my $EPP = DBI->connect("dbi:Pg:dbname=epp");
```

### Construct the Query

```perl
my $TRANSFERS = 5;
my $sql = q{SELECT t.owner_clid, t.transfer_to_clid, COUNT(t.*)
FROM transfer_request t
WHERE t.response LIKE '%%APPROVED'
GROUP BY t.transfer_to_clid, t.owner_clid
HAVING COUNT(t.*) > $TRANSFERS
ORDER BY t.owner_clid, t.transfer_to_clid;}
```

### Execute the Query

```perl
my $dbh = $EPP->prepare($sql);
$dbh->execute or die;
```
## .NA® Domain Name Transfers

Result of the Query, abbreviated ($TRANSFERS = 5) and sorted

<table>
<thead>
<tr>
<th>owner_clid</th>
<th>transfer_to_clid</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>afol</td>
<td>radonit</td>
<td>75</td>
</tr>
<tr>
<td>ascio</td>
<td>comlaude</td>
<td>6</td>
</tr>
<tr>
<td>dic</td>
<td>ihostnamibia</td>
<td>9</td>
</tr>
<tr>
<td>dic</td>
<td>verizon</td>
<td>10</td>
</tr>
<tr>
<td>dicf</td>
<td>verisign</td>
<td>8</td>
</tr>
<tr>
<td>escrow</td>
<td>dic</td>
<td>6</td>
</tr>
<tr>
<td>escrow</td>
<td>gijima</td>
<td>7</td>
</tr>
<tr>
<td>escrow</td>
<td>pyxis</td>
<td>64</td>
</tr>
<tr>
<td>escrow</td>
<td>swakopcom</td>
<td>21</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>verizon</td>
<td>intertech</td>
<td>6</td>
</tr>
<tr>
<td>verizon</td>
<td>itn</td>
<td>33</td>
</tr>
</tbody>
</table>
Fill up the Registrar hash, recycle colors, and 'draw' the 'edges'

```perl
while ( my @rec = $dbh->fetchrow_array ) {
    if ( $registrar ne $rec[0] ) {
        $farbe{$rec[0]} = $colors[$i];
        $registrar = $rec[0];

        if ( $i > $#colors ) {
            $i = 0;
        } else {
            $i ++;
        }
    }
    $graph->add_edge(from => $rec[0], to => $rec[1],
                     label => "$rec[2]",
                     color => $farbe{$rec[0]},
                     fontcolor => $farbe{$rec[0]});
}
```
'Draw’ all loosing Registrars from the Hash

```perl
foreach my $looser ( sort keys %farbe ) {
    $graph −> add_node(name => $looser, color => $farbe{$looser},
    fontcolor => $farbe{$looser});
}
```
'Draw' all loosing Registrars from the Hash

```perl
foreach my $looser ( sort keys %farbe ) {
    $graph −> add_node(name => $looser , color => $farbe{$looser},
    fontcolor => $farbe{$looser});
}
```

Run Graphviz, Writing the SVG file (here 't5.svg')

```perl
$graph −> run(format => 'svg', output_file => "t$TRANSFERS.svg");
```
Domain Name Transfers

Perl Code

'Draw' all loosing Registrars from the Hash

```perl
foreach my $looser ( sort keys %farbe ) {
    $graph -> add_node(name => $looser, color => $farbe{$looser},
                        fontcolor => $farbe{$looser});
}
```

Run Graphviz, Writing the SVG file (here 't5.svg')

```perl
$graph -> run(format => 'svg', output_file => "t$TRANSFERS.svg");
```

Write the DOT file (here 't5.dot') (Optional)

```perl
open(DOT, ">t$TRANSFERS.dot") or die;
print DOT $graph -> dot_input();
close DOT;
```
digraph Perl {
    node [fontsize="11" penwidth="3"]; 
    edge [fontsize="20" penwidth="3"]; 
    "afol" -> "radonit" [color="aquamarine" fontcolor="aquamarine" label="75"]; 
    "afol" [color="aquamarine" fontcolor="aquamarine"]; 
    "ascio" -> "comlaude" [color="blue" fontcolor="blue" label="6"]; 
    "ascio" [color="blue" fontcolor="blue"]; 
    "dic" -> "ihostnamibia" [color="blueviolet" fontcolor="blueviolet" label="9"]; 
    "dic" -> "verizon" [color="blueviolet" fontcolor="blueviolet" label="10"]; 
    "dic" [color="blueviolet" fontcolor="blueviolet"]; 
    "dicf" -> "verisign" [color="cadetblue" fontcolor="cadetblue" label="8"]; 
    "dicf" [color="cadetblue" fontcolor="cadetblue"]; 
    "escrow" -> "dic" [color="chartreuse" fontcolor="chartreuse" label="6"]; 
    "escrow" -> "gijima" [color="chartreuse" fontcolor="chartreuse" label="7"]; 
    "escrow" -> "pyxis" [color="chartreuse" fontcolor="chartreuse" label="64"]; 
    "escrow" -> "swakopcom" [color="chartreuse" fontcolor="chartreuse" label="21"]; 
    "escrow" [color="chartreuse" fontcolor="chartreuse"]; 
    
    "verizon" -> "intertech" [color="darkviolet" fontcolor="darkviolet" label="6"]; 
    "verizon" -> "itn" [color="darkviolet" fontcolor="darkviolet" label="33"]; 
    "verizon" -> "iway" [color="darkviolet" fontcolor="darkviolet" label="6"]; 
    "verizon" [color="darkviolet" fontcolor="darkviolet"]; 
}
.NA® Domain Name Transfers

More than 1 Transfer (from Foreign Registrar)
.NA® Domain Name Transfers
All Transfers (using circo)
## Transaction Patterns

### 24/7?

<table>
<thead>
<tr>
<th>Background</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Local Resellers Requested 24/7 Availability (2005)</td>
</tr>
<tr>
<td>- CoCCATools was deployed (2007)</td>
</tr>
<tr>
<td>- Resellers were promoted to Registrars (2008)</td>
</tr>
</tbody>
</table>
Transaction Patterns

24/7?

Background

- Local Resellers Requested 24/7 Availability (2005)
- CoCCATools was deployed (2007)
- Resellers were promoted to Registrars (2008)

Issue

Can Registrar Activity be displayed by the hour?
Background

- Local Resellers Requested 24/7 Availability (2005)
- CoCCATools was deployed (2007)
- Resellers were promoted to Registrars (2008)

Issue

Can Registrar Activity be displayed by the hour?

Tools Used

- R
  - RPostgreSQL
  - plotrix
- SSH Tunnel Manager
Set Up the Dates

mindate <- as.Date("2012-06-01")
maxdate <- as.Date("2013-05-31")
Set Up the Dates

```r
mindate <- as.Date("2012-06-01")
maxdate <- as.Date("2013-05-31")
```

Pull in the Packages

```r
require(RPostgreSQL)
require(plotrix)
```
Set Up the Dates

mindate <- as.Date("2012-06-01")
maxdate <- as.Date("2013-05-31")

Pull in the Packages

require(RPostgreSQL)
require(plotrix)

Configure the Driver and the Connection

drv <- dbDriver("PostgreSQL")
con <- dbConnect(drv, dbname = "epp")
Run the Query (for Nambian Registrars)

```r
hourNA <- dbGetQuery(con, paste("SELECT
    DATE_PART('hour', l.created) AS hour, COUNT(*) AS domains
FROM ledger l, domain d, client c
WHERE l.domain_name = d.name
    AND d.clid = c.clid AND c.country = 'NA'
    AND l.trans_type ilike 'Re%'
    AND DATE(l.created) BETWEEN ", mindate, ", maxdate, "'
GROUP BY DATE_PART('hour', l.created)
ORDER BY DATE_PART('hour', l.created) ASC
", sep = ""))
```

Transaction Patterns

R Code

Run the Query (for Namibian Registrars)

```r
hourNA <- dbGetQuery(con, paste("SELECT DATE_PART('hour',l.created) AS hour, COUNT(*) AS domains FROM ledger l, domain d, client c WHERE l.domain_name = d.name AND d.clid = c.clid AND c.country = 'NA' AND l.trans_type ilike 'Re%' AND DATE(l.created) BETWEEN " , mindate , " ' AND " , maxdate , " ' GROUP BY DATE_PART('hour',l.created) ORDER BY DATE_PART('hour',l.created) ASC ", sep = "" ))
```

Make Data Frames, count Transactions, set up Y-Axis

```r
hourlyNA <- data.frame(hour = hourNA$hour, domains = hourNA$domains)
hourlyFOR <- data.frame(hour = hourFOR$hour, domains = hourFOR$domains)
totNA <- sum(hourlyNA$domains)
totFOR <- sum(hourlyFOR$domains)
maxhoursNA <- round(max(hourlyNA$domains) + 50, -2)
```
Plot the Lines in Green

```r
plot(hourlyFOR$hour, hourlyFOR$domains, type = "l", col = "green",
     lwd = 5, ylab = "Domain Names", xlab = "", ylim = c(0, maxhoursNA),
     xlim = c(-0.1, 23.2), yaxs = "i", xaxs = "i", xaxt = "n", frame.plot = FALSE)
```

Plot the Bars in Blue

```r
lines(hourlyNA$hour, hourlyNA$domains, col = "blue", type = "h", lwd = 10)
```

Title, Axis, Legend

```r
title(xlab = "Time of Day")
axis(1, 0:23)
legend('topright', c(paste("Namibian Registrars : ", totNA),
                  paste("Foreign Registrars : ", totFOR)),
       col = c("blue", "green", 0),
       text.col = c("blue", "green", "black"), pch = NA, lwd = 4, cex = 0.7, bg = 0)
```
Transaction Patterns

R Code

**Plot the Lines in Green**

```r
plot(hourlyFOR$hour, hourlyFOR$domains, type = "l", col = "green",
     lwd = 5, ylab = "Domain Names", xlab = "", ylim = c(0, maxhoursNA),
     xlim = c(-0.1, 23.2), yaxs = "i", xaxs = "i", xaxt = "n", frame.plot = FALSE)
```

**Plot the Bars in Blue**

```r
lines(hourlyNA$hour, hourlyNA$domains, col = "blue", type = "h", lwd = 10)
```
Transaction Patterns

R Code

Plot the Lines in Green

```r
plot(hourlyFOR$hour, hourlyFOR$domains, type = "l", col = "green", lwd = 5, ylab = "Domain Names", xlab = ", ylim = c(0, maxhoursNA), xlim = c(-0.1, 23.2), yaxs = "i", xaxs = "i", xaxt = "n", frame.plot = FALSE)
```

Plot the Bars in Blue

```r
lines(hourlyNA$hour, hourlyNA$domains, col = "blue", type = "h", lwd = 10)
```

Title, Axis, Legend

```r
title(xlab = "Time of Day")
axis(1, 0:23)
legend("topright", c(paste("Namibian Registrars:", totNA), paste("Foreign Registrars:", totFOR)), col = c("blue", "green", 0), text.col = c("blue", "green", "black"), pch = NA, lwd = 4, cex = 0.7, bg = 0)
```
Transaction Patterns

Results (not 24/7)
What is the ratio between 2\textsuperscript{nd} vs 3\textsuperscript{rd} level domain names?
Registration Patterns

Issue

What is the ratio between 2\textsuperscript{nd} vs 3\textsuperscript{rd} level domain names?

Run the query

```r
dpct <- dbGetQuery(con, paste("SELECT DISTINCT UPPER(d.zone) AS zone, COUNT(d.*) AS domains, ROUND(COUNT(d.*) * 100.0/(SELECT COUNT(*) FROM domain), 2) AS percent FROM domain d, ledger l WHERE l.domain_name = d.name GROUP BY zone ORDER BY percent DESC"))
```

Date Constraint not displayed
### Issue

What is the ratio between 2\textsuperscript{nd} vs 3\textsuperscript{rd} level domain names?

### Run the query

```r
dpct <- dbGetQuery(con, paste("SELECT
    DISTINCT UPPER(d.zone) AS zone,
    COUNT(d.*) AS domains,
    ROUND(COUNT(d.*) * 100.0/(SELECT COUNT(*) FROM domain), 2) AS percent
FROM domain d, ledger l
WHERE l.domain_name = d.name
GROUP BY zone
ORDER BY percent DESC")
```

Date Constraint not displayed

### Draw the Chart

```r
pie3D(dpct$domains, labels = paste(dpct$zone, dpct$percent, sep=" "),
      col = c(2,3,4,5,6,7,8,9,10), labelcol = "black", explode = 0,
      labelcex = 0.6, height = 0.15, radius = 1)
```

Title and Legend not displayed
## Registration Patterns

Result of the Query

<table>
<thead>
<tr>
<th>zone</th>
<th>percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM.NA</td>
<td>72.01</td>
</tr>
<tr>
<td>CO.NA</td>
<td>12.89</td>
</tr>
<tr>
<td>NA</td>
<td>9.28</td>
</tr>
<tr>
<td>ORG.NA</td>
<td>4.40</td>
</tr>
<tr>
<td>EDU.NA</td>
<td>1.26</td>
</tr>
<tr>
<td>NET.NA</td>
<td>0.16</td>
</tr>
</tbody>
</table>
Can be done on FRED and CoCCATools

- On any SQL database, in fact
- Any Aggregate (GROUP BY, COUNT, etc) SQL Query can be graphed
- SELECT is safe

Automatic Report Generation

- SWEAVE
  - \LaTeX
  - RGraphviz

Report / Presentation Writing

- LyX/\LaTeX
  - Beamer
  - (SWEAVE)

- Open Office?

Kolophonium

- Open Source
- Quality Output
Discussion

Much More is Possible

- Can be done on FRED and CoCCATools
  - On any SQL database, in fact
  - Any Aggregate (GROUP BY, COUNT, etc) SQL Query can be graphed
  - SELECT is safe

- Automatic Report Generation
  - SWEAVE
    - \LaTeX
    - RGraphviz

- Report / Presentation Writing
  - \textsc{LyX/\LaTeX}
    - Beamer
    - (SWEAVE)
  - Open Office?

- Kolophonium
  - Open Source
  - Quality Output
Discussion

Much More is Possible

- Can be done on FRED and CoCCATools
  - On any SQL database, in fact
  - Any Aggregate (GROUP BY, COUNT, etc) SQL Query can be graphed
  - SELECT is safe

- Automatic Report Generation
  - SWEAVE
    - LATEX
    - RGraphviz

- Report / Presentation Writing
  - LyX/LATEX
    - Beamer
    - (SWEAVE)

- Kolophonium
  - Open Source
  - Quality Output
Discussion
Much More is Possible

- Can be done on FRED and CoCCATools
  - On any SQL database, in fact
  - Any Aggregate (GROUP BY, COUNT, etc) SQL Query can be graphed
  - SELECT is safe

- Automatic Report Generation
  - SWEAVE
    - \LaTeX
    - RGraphviz

- Report / Presentation Writing
  - LyX/\LaTeX
    - Beamer
    - (SWEAVE)
  - Open Office?

- Kolophonium
  - Open Source
  - Quality Output