Mapping election results with CartoWeb
Mapping election results with CartoWeb

Plan

- Introduction
  - Project presentation
  - Constraints
  - Slide show
  - Processes
- Pseudo-dynamic Mapfile generation:
  - Automatic generation of multiple layers
  - Automatic adaptation of class intervals
- Load / performance tuning for high traffic:
  - Caching images with CartoWeb
  - Database tuning
- Access statistics
- Possible improvements
- Questions
The goal of this web mapping project is:
- to show election results live using thematic maps. This includes evolution maps between 2001 & 2006.

Constraints:
- No Plugin for the client -> raster image
- Open source software to adapt it to our needs
- Human intervention on the election day should be limited to acquiring the data
- MicroGIS Know-how

Software's Choice
- Mapserver/CartoWeb in relation with PostgreSQL/PostGIS
Introduction

Slide show: thematics
Introduction

Slide show: thematics
Introduction

Slide show: thematics
Introduction

Slide show: thematics
Introduction

Slide show: functionalities
Introduction

Slide show: functionalities
Introduction

Slide show: functionalities
Introduction

Slide show: functionalities
Introduction

Processes

• Based on CartoWeb/PostGIS, 4 steps are needed to generate the maps:
  - AQUISITION
    - Copy&Paste
  - STRUCTURING
    - PL/pgSQL scripting
    - Materialized views
  - CLASSIFICATION
    - autolayers.php
    - SQL requests
    - makemaps.php
    - Mapfile templates
  - PRESENTATION
    - IMG Mapserver
    - GUI CartoWeb

• Generated: 28 thematics; every thematic is automatically divided into 5 class intervals
• The project has to automate these processes and to customize them to serve high traffic.
Pseudo-dynamic mapfile generation

Automatic generation of multiple layers

- The **autolayer** functionality from CartoWeb allows to generate multiple layers from a template without any notion of PHP.
Pseudo-dynamic mapfile generation

Automatic adaptation of class intervals

- **makemaps.php** is a script of CartoWeb that allows to parse a PHP template to generate a mapfile.

  - `<php
  
  • SELECT min(colname), max(colname) FROM layer;
  
  • MyFunction computes class intervals: min1, max1, min2, max2, etc
  
  • print_f to MAPFILE
    
      - EXPRESSION (([colname] > min1) AND ([colname] <= max1))
      
      - EXPRESSION (([colname] > min2) AND ([colname] <= max2))
      
      - ...

  `>

- Whenever new values are added to the database, makemaps.php should be invoked to generate the new adjusted mapfile.

  - `<PHP-INTERPRETER> makemaps.php`
Orientation

CLASSIFICATION: pseudo-dynamic Mapfile generation

PRESENTATION: caching images with CartoWeb
• Introducing some facts:
  − Mapserver generates an image on request. A list of parameters describes the map.
  − The image is associated to a pseudo unique filename and returned to the browser.

=> Each request, even with identical parameters generates a new image/filename.

• CartoWeb offers a caching mechanism. If the same parameters are invoked, then the cached image is sent again.

=> This is a great functionality, but how can you use it?
To request a cached image, the new request should contain the same parameters as the cached request.

Which parameters are commonly used in a project?
- Mapped layers, Map size & Extent (Zoom&Pan)

We have an infinite quantity of possible maps, because of free zoom and free pan possibilities.

For the cache to work, we have to reduce the amount of different maps:
- 1/ disable free pan
- 2/ disable free zoom-in, zoom-out

And to preserve interaction, we have substituted these functionalities with a drop-down menu to access predefined areas. We reduced the amount of maps to 1120 (14 thematics * 2 years * 20 predefined areas * 2 image sizes).

Whenever the database id updated, the cache should be cleaned:
- `<PHP-INTERPRETER> cw3setup.php --clean`
PRESENTATION: caching images with CartoWeb

STRUCTURING: database tuning
Access time to views is a bottleneck => need to optimization
- Select operations have to be kept simple!
Materialized views used by the application.

- No join between tables is required.
- A PL/pgSQL script generates these tables from previous structure.
The website is unknown before election day.
On the election day, the following websites have linked to our site:
- TSR.ch (state TV)
- RSR.ch (state Radio)
- LeTemps.ch (major newspaper)
- RougeFM.ch (local radio)

Traffic summary:

<table>
<thead>
<tr>
<th></th>
<th>Hits</th>
<th>Visitors</th>
<th>Pages Views</th>
<th>Bandwidth</th>
<th>Visit length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunday</td>
<td>1,614,810</td>
<td>5,518</td>
<td>50,333</td>
<td>6.9 GB</td>
<td>10:08</td>
</tr>
<tr>
<td>Monday</td>
<td>664,277</td>
<td>3,079</td>
<td>19,834</td>
<td>2.8 GB</td>
<td>6:58</td>
</tr>
</tbody>
</table>

These figures are equivalent on the Swiss scale to:
- 17. position in Pages Views
- 27. position in Visitors

(source: www.wemf.ch)
Access statistics

Activity by Hour of Day

- Offices open Monday morning
- Results published
- Dinner & TV evening news

Time range: 12.03.2006 12:00:16 – 13.03.2006 11:59:59
Conclusions

Possible improvements

• The webmapping software package CartoWeb/PostGIS is adequate for thematic cartography of news events on a regional basis.

• For larger diffusion, some points should be further evaluated:
  - Reduction of bandwidth use
  - Use of Apache caching mechanism
  - Deployment of a PostgreSQL/PostGIS caching mechanism for queries (info tool)
  - Improvement of CartoWeb caching mechanism: limitation of parallel maps generation

• Richer thematic maps could be implemented (circle proportional, pie charts, etc): see Mapserver or CartoWeb functionalities.
Thank you for your attention!