MapProxy

The caching proxy for web map services

Oliver Tonnhofer <olt@omniscale.de>
From Oldenburg in Germany

We focus on:

- OpenSource WebGIS, Server development
- OpenStreetMap WMS
- MapProxy development and support
Content

What is it?

About the project

Examples

News
What is it?
Clients

MapProxy

Server
Cache

Improve WMS Performance
Tiled access

Change
Protocols/versions
Image formats
Projection

Enhance
Watermarks
Attribution lines
Combine Layer/Server
## Sources

<table>
<thead>
<tr>
<th>WMS</th>
<th>WMS-C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0/1.1.1/1.3.0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TMS</th>
<th>Custom Tiles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Google Maps/Bing/etc.)</td>
</tr>
</tbody>
</table>
Services

<table>
<thead>
<tr>
<th>WMS</th>
<th>WMS-C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0/1.1.1/1.3.0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TMS</th>
<th>KML</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Super Overlays</td>
</tr>
</tbody>
</table>

About the project
History

Started late 2008

In production since mid 2009

OpenSource since early 2010
Releases

0.8.4 current

0.9.0 upcoming release
<table>
<thead>
<tr>
<th>Written in</th>
<th>Python</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runs on</td>
<td>Linux/Unix</td>
</tr>
<tr>
<td></td>
<td>Windows</td>
</tr>
<tr>
<td>License</td>
<td>Affero GPL v3</td>
</tr>
<tr>
<td>Tested</td>
<td>lots of unit/system tests</td>
</tr>
<tr>
<td>Deployment</td>
<td>HTTP or FastCGI</td>
</tr>
<tr>
<td>Feature</td>
<td>Available</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Distributed SCM</td>
<td>✓</td>
</tr>
<tr>
<td>Issue tracker</td>
<td>✓</td>
</tr>
<tr>
<td>Mailing list</td>
<td>✓</td>
</tr>
<tr>
<td>Blog</td>
<td>✓</td>
</tr>
</tbody>
</table>
Example-driven introduction
Getting started

% paster create --template \ mapproxy_conf foss4g_example
% cd foss4g_example
% paster serve etc/develop.ini --reload
http://localhost:8080/service?...
Example 1

Enhance existing WMS server

<table>
<thead>
<tr>
<th>BlueMarble</th>
<th>Satellite imagery</th>
<th>Background</th>
</tr>
</thead>
<tbody>
<tr>
<td>by NASA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>US population</th>
<th>Vector data</th>
<th>Overlay</th>
</tr>
</thead>
<tbody>
<tr>
<td>by demo.opengeo.org</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Map showing US population data overlayed on satellite imagery]
sources

sources:
  us_pop_wms:
    type: wms
    req:
      url: http://demo.opengeeo.org/geoserver/wms?
      layers: ‘topp:states’
      transparent: true
      format: image/png
  bluemarble_wms:
    type: wms
    supported_srs: ['EPSG:4326']
    req:
      url: http://neowms.sci.gsfc.nasa.gov/wms/wms?
      layers: BlueMarbleNG
caches & layers

caches:
    us_pop_cache:
      sources: [bluemarble_wms, us_pop_wms]
      grids: [GLOBAL_GEODEETIC]

layers:
  world:
    title: BlueMarble + US Population
    sources: [us_pop_cache]
services:
  tms:
  kml:
  wms:
    srs: ['EPSG:900913', 'EPSG:4326']
    image_formats: [image/png, image/jpeg]
  md:
    title: MapProxy WMS Demo
    abstract: FOSS4G Demo
    contact:
      person: Oliver Tonnhofer
      organization: Omniscale
WMS

GetCapabilities

http://localhost:8080/service?request=GetCapabilities&service=WMS

GetMap

EPSG:4326  EPSG:900913
http://localhost:8080/tms/1.0.0/world_EPSG4326

<?xml version="1.0" encoding="UTF-8" ?>
<TileMap version="1.0.0">
  <Title>BlueMarble + Country Borders</Title>
  <Abstract></Abstract>
  <SRS>EPSG:4326</SRS>
  <BoundingBox minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0" />
  <Origin x="-180.0" y="-90.0" />
  <TileFormat width="256" height="256" mime-type="image/jpeg" extension="jpeg" />
  <TileSets profile="global-geodetic">
    <TileSet href="http://localhost:8080/tms/1.0.0/world_EPSG4326/0"
             units-per-pixel="0.703125" order="0" />
    <TileSet href="http://localhost:8080/tms/1.0.0/world_EPSG4326/1"
             units-per-pixel="0.3515625" order="1" />
    ...
  </TileSets>
</TileMap>
TMS

http://localhost:8080/tms/world_EPSG4326/3/4/5.jpeg

HTTP cache control

ETag: 5c4a1f9bca1b95b283a512357ff42fc
Cache-control: max-age=259200 public ...

HTTP/1.1 304 Not Modified
TMS

http://localhost:8080/tms/world_EPSG4326/3/4/5.jpeg
KML

http://localhost:8080/kml/world_EPSG4326/0/0/0.kml
Spherical mercator?
Google Maps, Bing, OSM, etc.

caches:
  us_pop_cache:
    sources: [bluemarble_wms, us_pop_wms]
    grids: [GLOBAL_GEODETiC, GLOBAL_MERCATOR]

Reprojects BlueMarble from EPSG:4326
adds tms/1.0.0/world_EPSG900913
Featureinfo?

sources:
  us_pop_wms:
    type: wms
    wms_opts:
      featureinfo: true
  req:
    url: http://demo.opengeo.org/geoserver/wms?
    layers: 'topp:states'
    transparent: true
    format: image/png

world layer is now queryable
GetFeatureInfo
Example 2

Improve WMS performance

OpenStreetMap WMS with CPU/IO intensive rendering

- Large dataset
- Lots of details
- Rendering with anti-aliasing
sources:

osm_wms:
  type: wms
  req:
    url: http://localhost:81/mapnik?
    layers: osm
caches:
  osm_cache:
    sources: [osm_wms]
    grids: [osm_germany]
    link_single_color_images: true
Caching

MapProxy allows free-zooming but caches at fixed resolutions
Vector scaling
Vector scaling
grids:
  osm_germany:
    srs: 'EPSG:25832'
    bbox: [-3.82, 46.18, 24.91, 55.23]
    bbox_srs: 'EPSG:4326'
    res_factor: sqrt2
More grid options

<table>
<thead>
<tr>
<th>min_resolution</th>
<th>max_resolution</th>
<th>num_levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>tile_size</td>
<td>res</td>
<td>res_factor</td>
</tr>
</tbody>
</table>
Cache pre-generation

% mapproxy-seed -f etc/mapproxy.yaml \ etc/seed.yaml -c 8
Cache pre-generation

% mapproxy-seed -f etc/mapproxy.yaml \\ etc/seed.yaml -c 8

Multiprocessing
seed.yaml

views:
  germany:
    bbox: [7.36, 46.33, 14.28, 56.09]
    bbox_srs: 'EPSG:4326'
    level: [0, 14]
    srs: ['EPSG:900913', 'EPSG:4326']

seeds:
  osm:
    views: ['germany']
    remove_before:
      month: 1
      days: 3
Any OGR source (Shapefile, PostGIS, etc.)

germany:
  ogr_datasource: 'shps/world_boundaries_m.shp'
  ogr_where: 'CNTRY_NAME = "Germany"'
  ogr_srs: 'EPSG:900913'
  level: [0, 14]
  srs: ['EPSG:900913', 'EPSG:4326']

WKT polygons  http://mapproxy.org/static/polygons/

germany:
  polygons: 'polygons/GM.txt'
  polygons_srs: EPSG:900913
  level: [0, 14]
  srs: ['EPSG:900913', 'EPSG:4326']
Seed strategy

Normal

level per level, top to bottom, left to right
Seed strategy

Normal

level per level, top to bottom, left to right

mapproxy-seed

recursive depth-first traversal

works with the OS/DB cache
PNG8 870x530 = 50ms
Reprojected
PNG8 870x530 = 130ms
Single Tile < 1ms
News
Python Image Library

Really fast C-based imaging library

only Issues: slow PNG performance no transparency PNG8
Python Image Library

Really fast C-based imaging library

Now fixed! 10x faster – same as libjpeg
full transparency for PNG8

to be included in next PIL release

http://bitbucket.org/olt/pil-117-fastpng
Jython

Python implementation for JVM
no support for C extensions
Jython

Python implementation for JVM
no support for C extensions

Jython wrapper with
Java ImageIO/JAI
GeoTools
Jython

Python implementation for JVM
no support for C extensions

Jython wrapper with
Java ImageIO/JAI
GeoTools

MapProxy as .war
alpha release
Thank you.

Oliver Tonnhofer <olt@omniscale.de>

Visit http://mapproxy.org
Join http://lists.osgeo.org/mailman/listinfo/mapproxy
Read http://blog.mapproxy.org
Backup tiles below
Level 3

Level 4
Seed with avg. 150 tiles/second on a Quad Core (i7@2.6GHz)
## Deployment

<table>
<thead>
<tr>
<th></th>
<th>Develop</th>
<th>Production</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Protocol</strong></td>
<td>HTTP</td>
<td>FastCGI or HTTP</td>
</tr>
<tr>
<td><strong>Server</strong></td>
<td>embedded</td>
<td>Apache, nginx, lighttpd, varnish, squid, ...</td>
</tr>
</tbody>
</table>

*with auto reloading*