# Vector tile benchmark

Pirmin Kalberer, @implgeo
Sourcepole, Switzerland

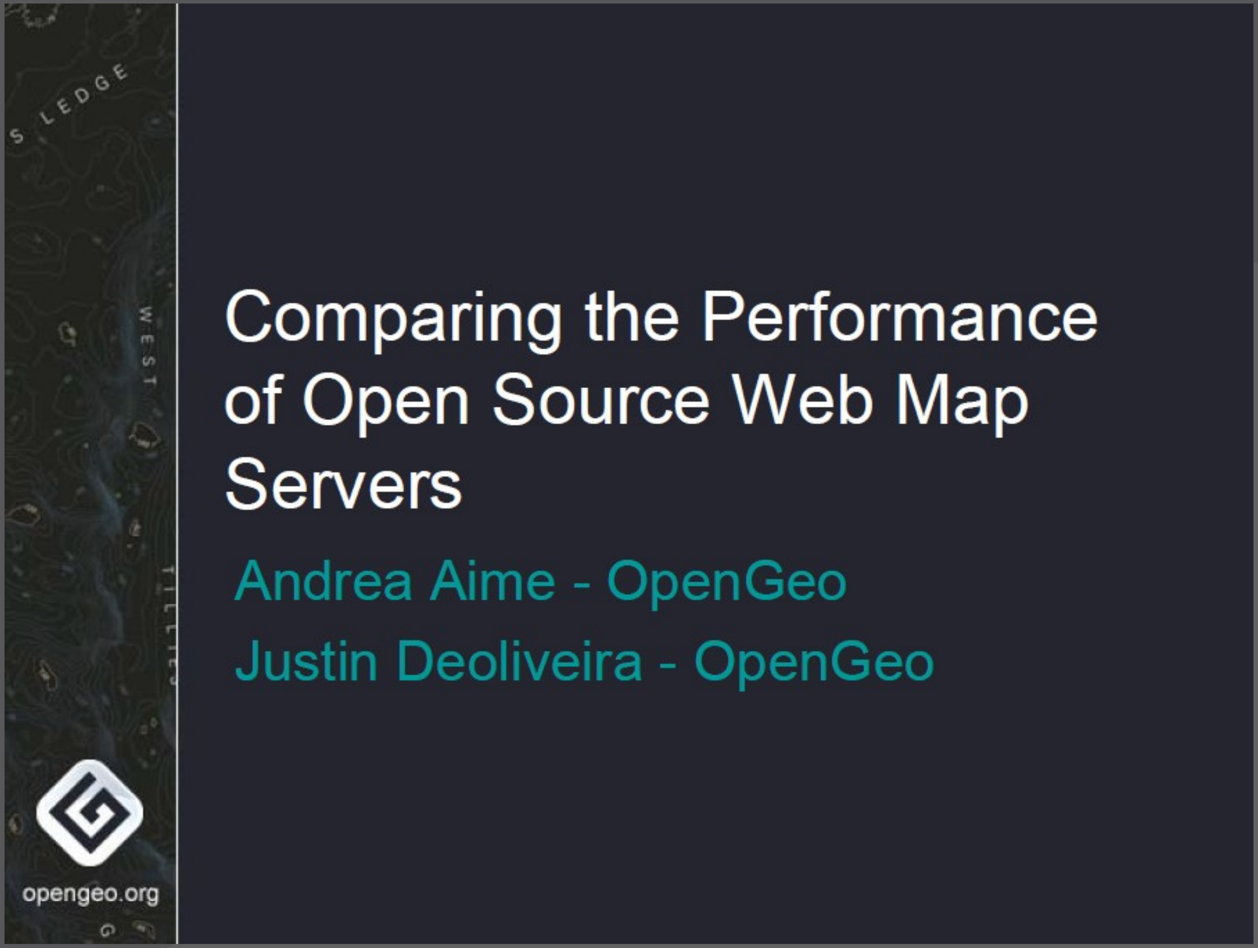
Karsten Vennemann

TERRA GIS
Terrestrial EnviRonment Regional Analysis

SOURCEPOLE
Linux & Open Source Solutions

# FOSS4G benchmark tradition

- WMS benchmark 2008



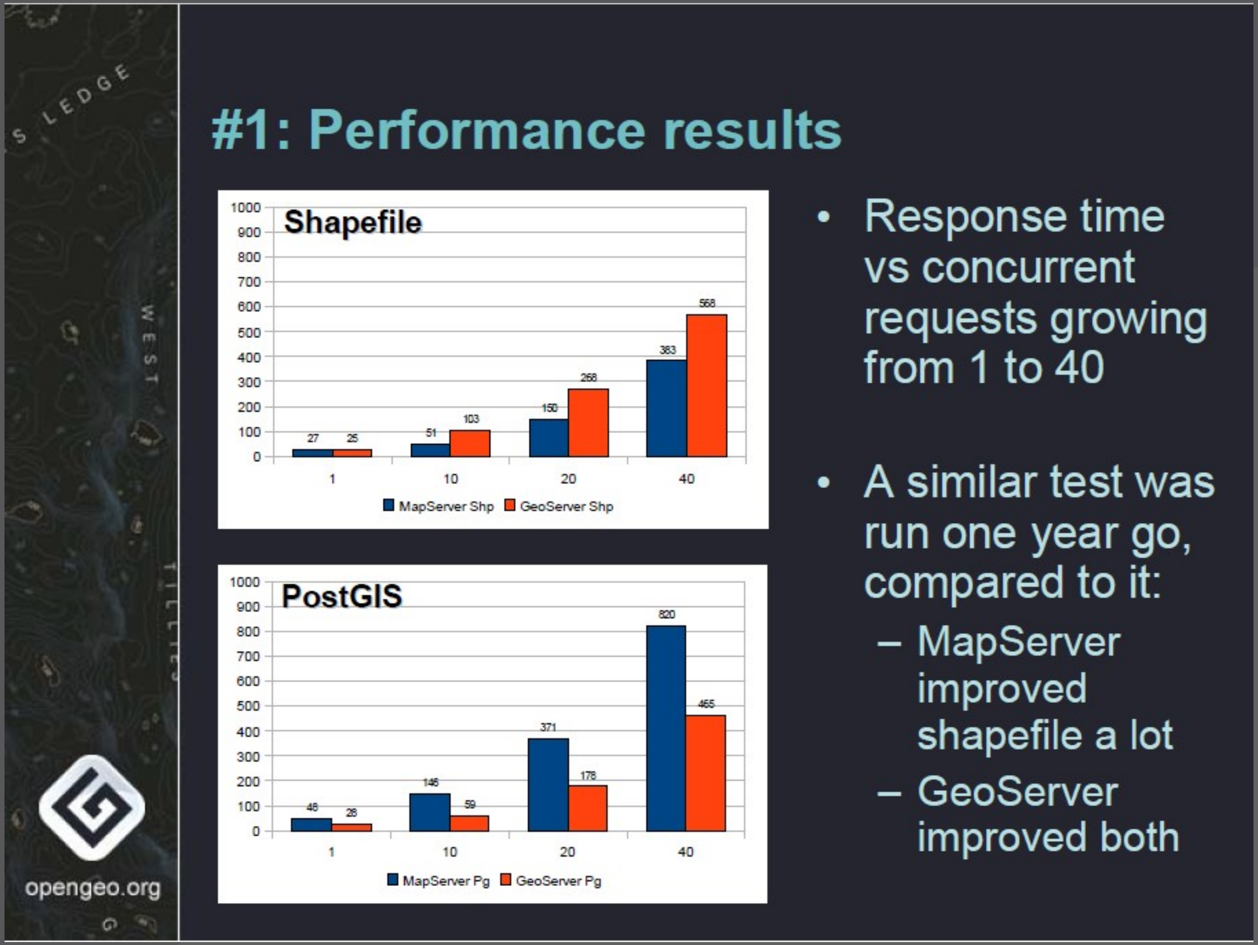
Comparing the Performance of Open Source Web Map Servers

Andrea Aime - OpenGeo

Justin Deoliveira - OpenGeo

# FOSS4G benchmark tradition

- WMS benchmark 2008

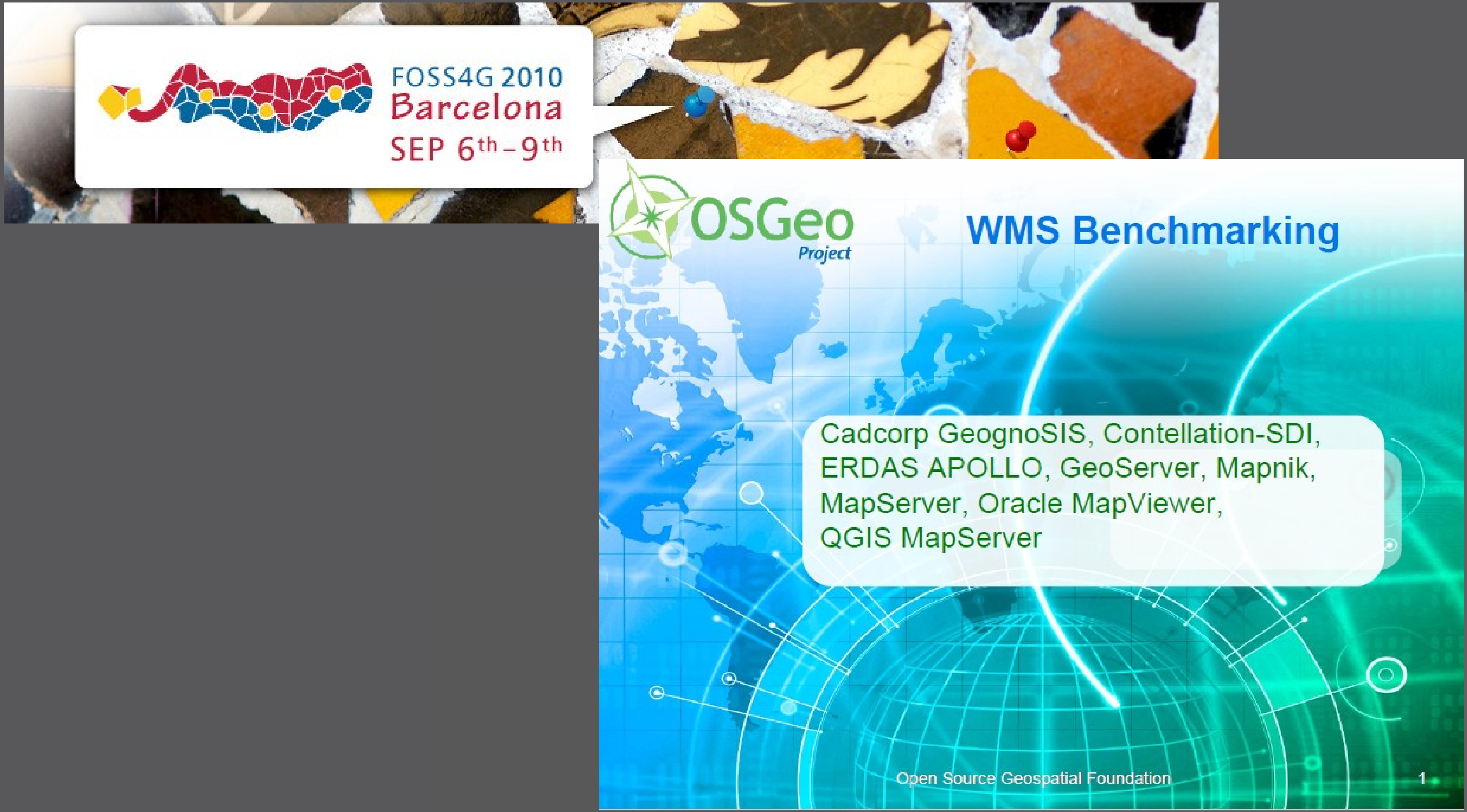


## #1: Performance results

- Response time vs concurrent requests growing from 1 to 40
- A similar test was run one year go, compared to it:
  - MapServer improved shapefile a lot
  - GeoServer improved both

# FOSS4G benchmark tradition

- WMS benchmark 2010



WMS Benchmarking

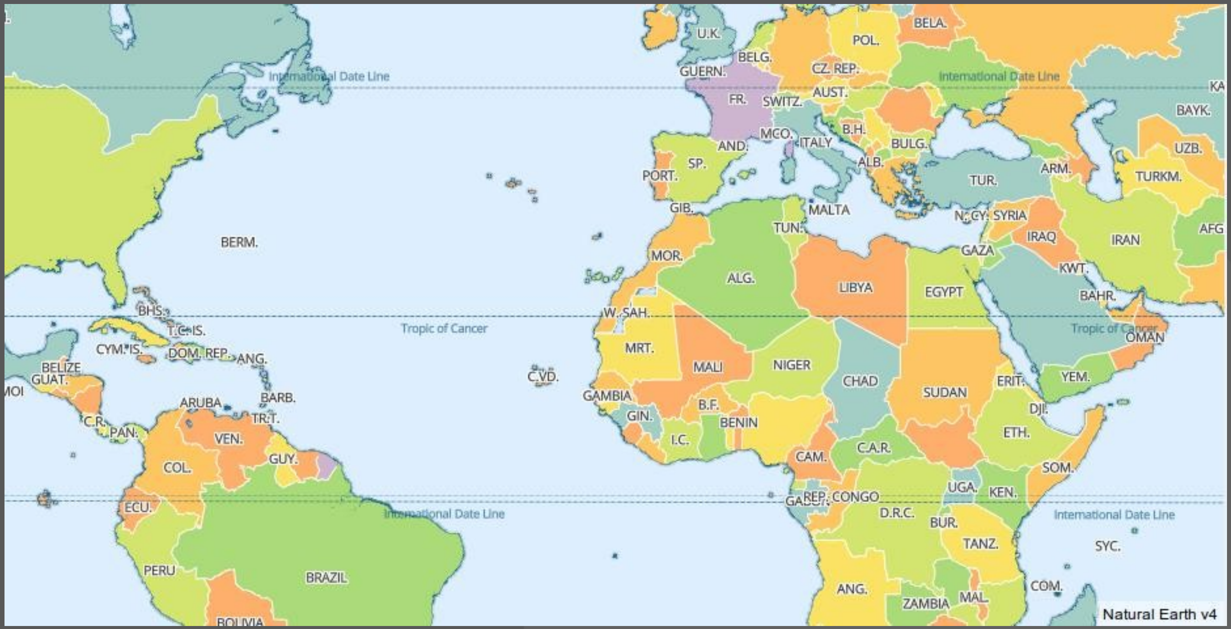
Cadcorp GeognoSIS, Contellation-SDI, ERDAS APOLLO, GeoServer, Mapnik, MapServer, Oracle MapViewer, QGIS MapServer

Open Source Geospatial Foundation

1

# A vector tile benchmark

- Natural Earth dataset
- Data stored in PostGIS
- Test style:



- https://github.com/pka/mvt-benchmark

# Tileset definition

- Tileset name: `ne_countries`
- Maxzoom level for tile data: 6
- Tile size: 4096
- SRS (data and tiles): EPSG:3857 (Web Mercator)

# Layer definition

| Name | Geom. type | Buffer | Simplify | Table | Attributes | Conditions |
|---|---|---|---|---|---|---|
| country | MULTIPOLYGON | 3 | yes | ne_10m_admin_0_countries | adm0_a3, mapcolor7 | min_zoom <= {z} |
| country-name | POINT | 0 | no | ne_10m_admin_0_country_points | abbrev, name | - |
| geo-lines (z=1..4) | MULTILINESTRING | 0 | no | ne_50m_geographic_lines | name | - |
| geo-lines (z=5..6) | MULTILINESTRING | 0 | no | ne_10m_geographic_lines | name | - |
| land-border-country | MULTILINESTRING | 0 | yes | ne_10m_admin_0_boundary_lines_land | - | min_zoom <= {z} |
| state | MULTILINESTRING | 0 | yes | ne_10m_admin_1_states_provinces_lines | adm0_a3 | min_zoom <= {z} |

# Benchmark measurment #1

- How long does it take to generate all tiles (single node / multiple nodes)

# Vector tile creation

- Read geodata within tiles borders
- Clip geometries
- Simplify geometries
  - Polygons: e.g. SnapToGrid
  - Lines: e.g. Douglas-Peucker
- (Generate label points)
- Deliver MVT (Protobuf) format

# Benchmark measurment #2

- How many requests/s does the tile server deliver in web server mode

# Serving vector tiles

- **MVT creation on request with file cache (capabable of delivering live data)**
- **Not part of benchmark:**
  - Serving static MVT files directly
    - Webserver (Apache, Nginx)
    - Service: S3, etc.
  - Cache-, CORS- headers
  - Layer concatenation
- **Measured with wrk, a high performance http load testing tool**

# Contestants 2019

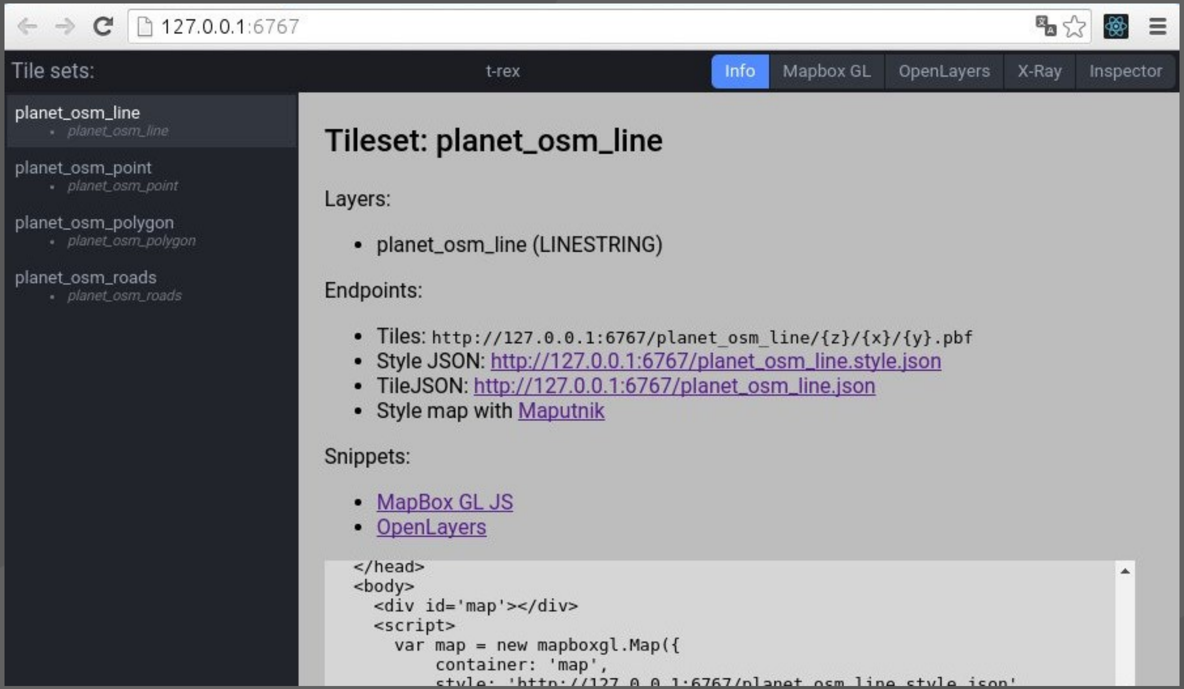
- T-rex
- UMN Mapserver

More tile servers:

https://github.com/mapbox/awesome-vector-tiles

# t-rex

- **Serve vector tiles**
  - Live tiles from PostGIS databases and GDAL vector formats
  - Zero-configuration mode
  - Embedded webserver
  - Visual styling with Maputnik



Tileset: planet_osm_line

Layers:

- planet_osm_line (LINESTRING)

Endpoints:

- Tiles: `http://127.0.0.1:6767/planet_osm_line/{z}/{x}/{y}.pbf`
- Style JSON: http://127.0.0.1:6767/planet_osm_line.style.json
- TileJSON: http://127.0.0.1:6767/planet_osm_line.json
- Style map with Maputnik

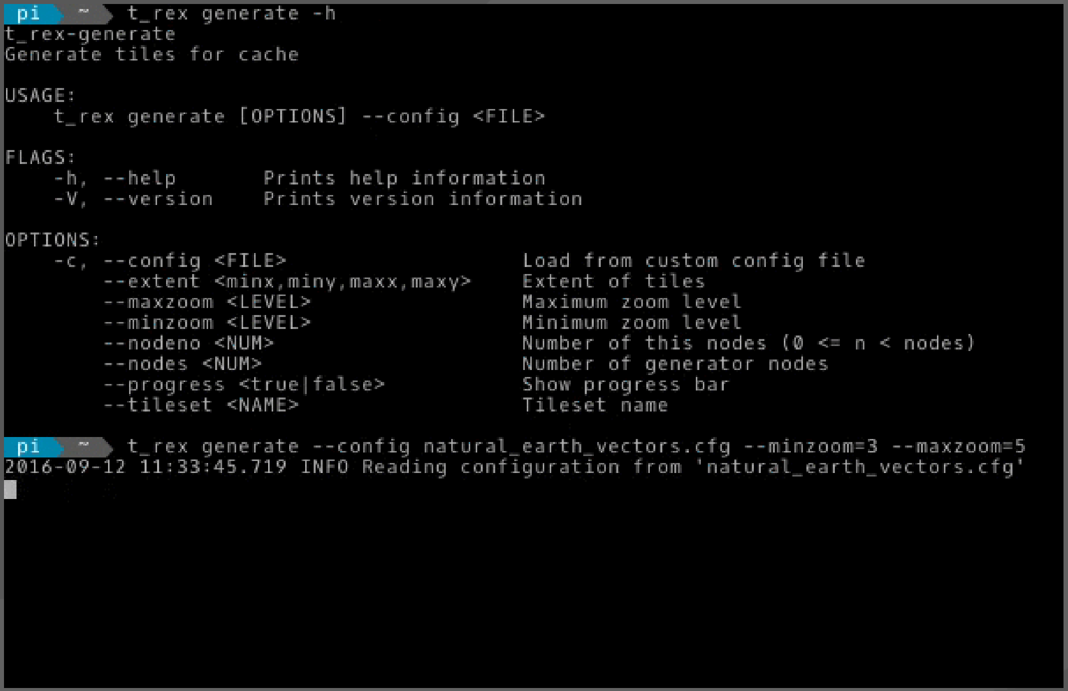
Snippets:

- MapBox GL JS
- OpenLayers

```
  </head>
  <body>
    <div id='map'></div>
    <script>
      var map = new mapboxgl.Map({
          container: 'map',
          style: 'http://127.0.0.1:6767/planet_osm_line.style.json',
```

# t-rex

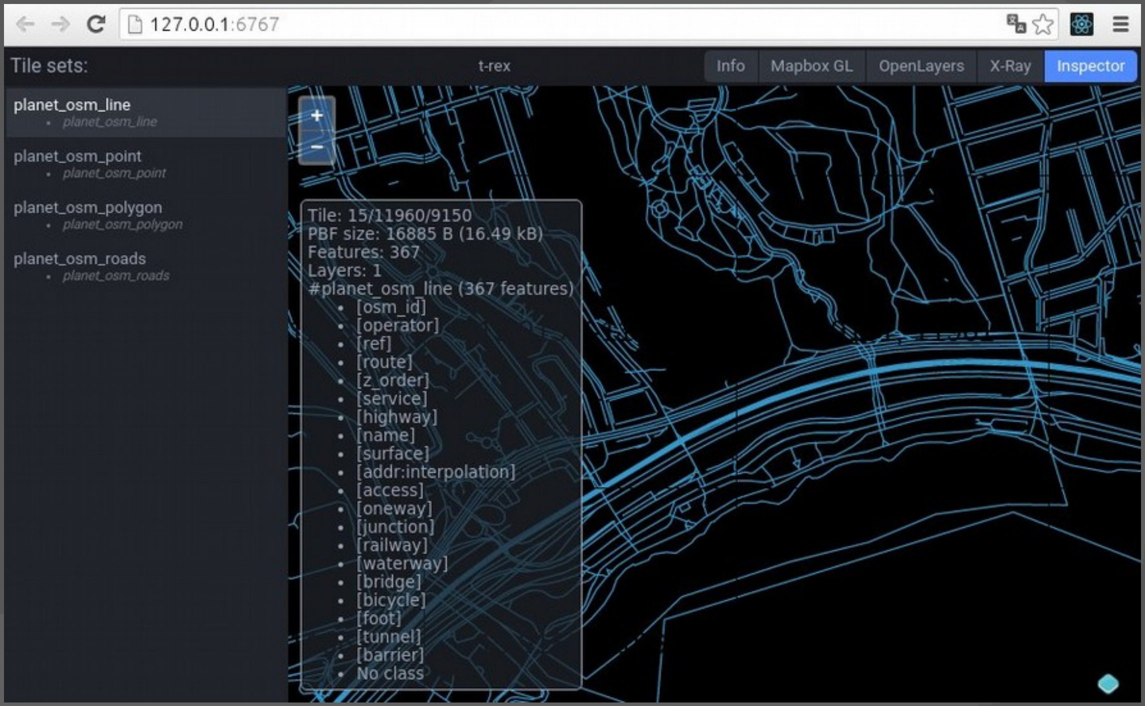
- **Generate vector tiles**
  - Tile generation command with simple parallelization
  - Generate configuration template
  - Support for custom tile grids



```
 pi  ~  t_rex generate -h
t_rex-generate
Generate tiles for cache

USAGE:
    t_rex generate [OPTIONS] --config <FILE>

FLAGS:
    -h, --help       Prints help information
    -V, --version    Prints version information

OPTIONS:
    -c, --config <FILE>                        Load from custom config file
        --extent <minx,miny,maxx,maxy>         Extent of tiles
        --maxzoom <LEVEL>                      Maximum zoom level
        --minzoom <LEVEL>                      Minimum zoom level
        --nodeno <NUM>                         Number of this nodes (0 <= n < nodes)
        --nodes <NUM>                          Number of generator nodes
        --progress <true|false>                Show progress bar
        --tileset <NAME>                       Tileset name

 pi  ~  t_rex generate --config natural_earth_vectors.cfg --minzoom=3 --maxzoom=5
2016-09-12 11:33:45.719 INFO Reading configuration from 'natural_earth_vectors.cfg'
```

# t-rex

- **Easy to Use**
  - Auto-detection of layers in database
  - Single human readeable configuration file
  - Automatic reprojection to grid CRS
  - Install package or run in a Docker container



```
Tile: 15/11960/9150
PBF size: 16885 B (16.49 kB)
Features: 367
Layers: 1
#planet_osm_line (367 features)
  • [osm_id]
  • [operator]
  • [ref]
  • [route]
  • [z_order]
  • [service]
  • [highway]
  • [name]
  • [surface]
  • [addr:interpolation]
  • [access]
  • [oneway]
  • [junction]
  • [railway]
  • [waterway]
  • [bridge]
  • [bicycle]
  • [foot]
  • [tunnel]
  • [barrier]
  • No class
```

# MapServer

- **Originally developed at the University of Minnesota (UMN), short "MapServer"**
  - one of the most mature open source projects (1994)
  - written in C
- **Main Focus**
  - rendering spatial data
  - development environment for spatially-enabled internet applications

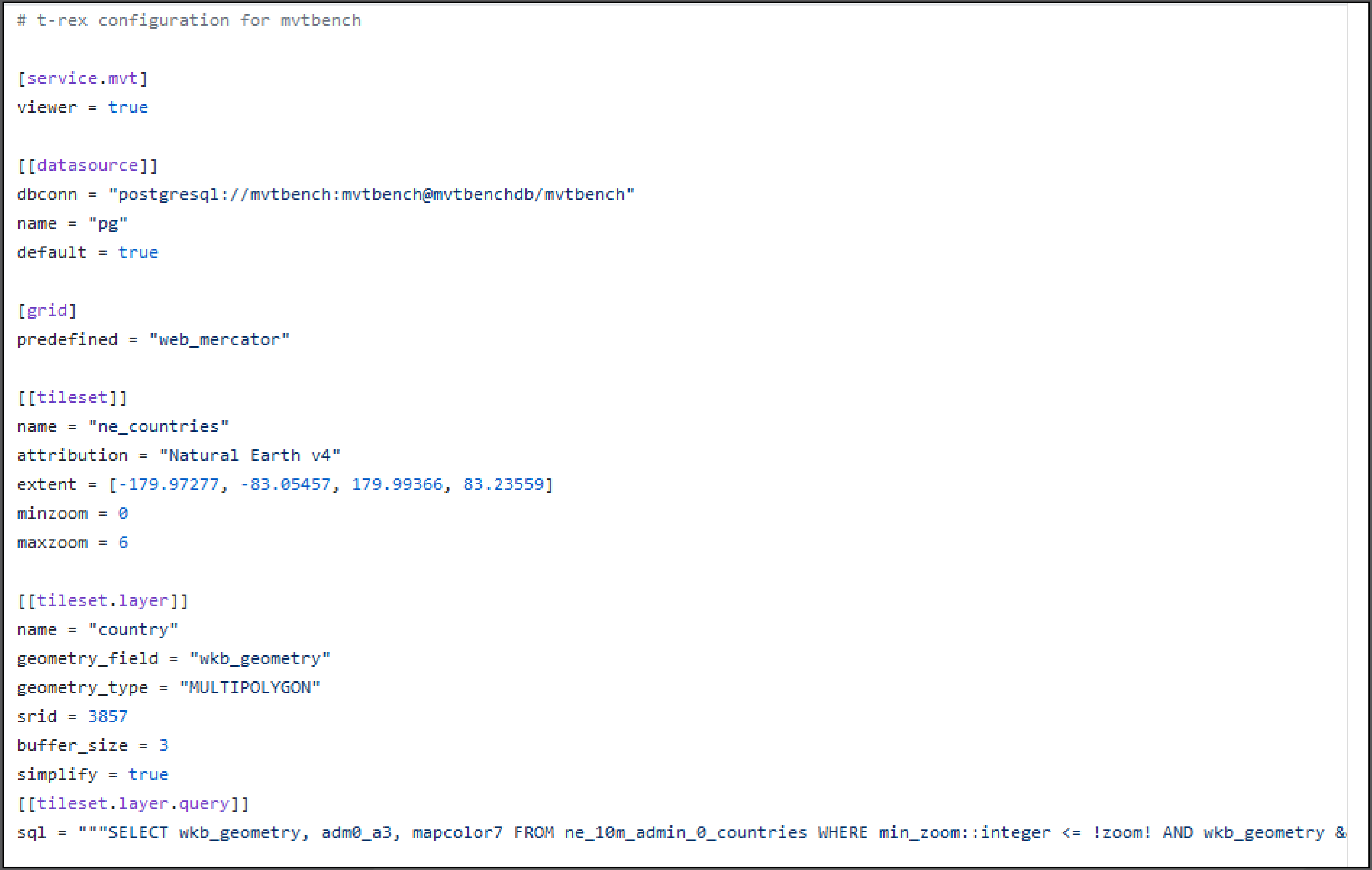
# MapServer

- **Map output**
  - CGI mapserv (Linux) and mapserv.exe (windows)
  - MapScript API available for Python, PHP, Perl, and Java
  - Map/Layer configuration text file .map
- **Formats**
  - In: PostGIS, Oracle Spatial ArcSDE, WMS, GDAL and OGR formats
  - Out: GIF, JPG, PNG, all GDAL formats, WFS and WMS

# Configuration for benchmark data

- T-rex: TOML with embedded SQL
- MapServer: Map-File with embedded SQL + XML-File for MapCache

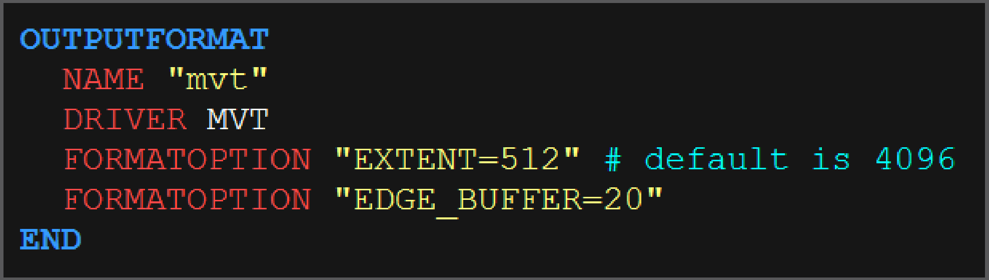
# t-rex



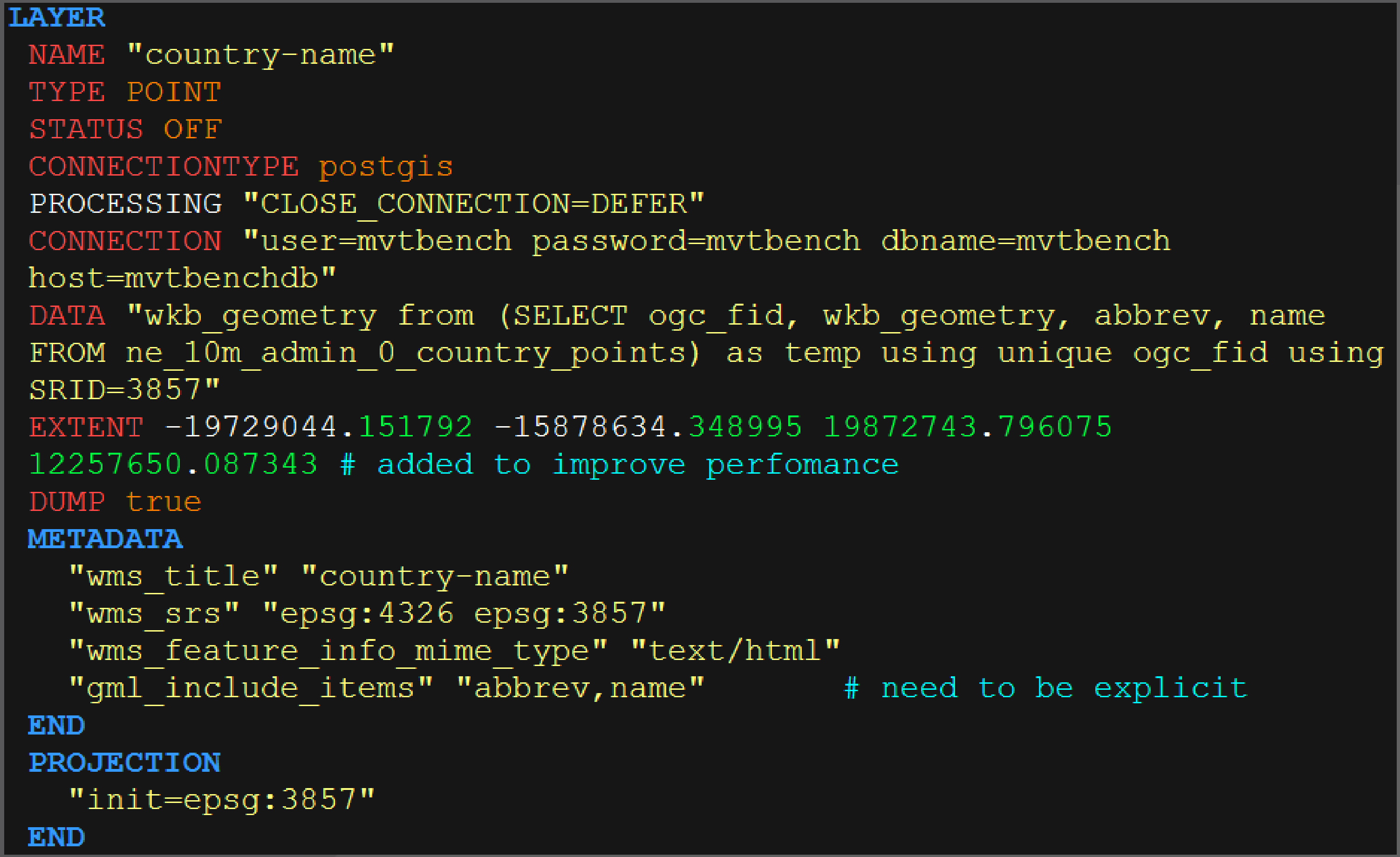
```
# t-rex configuration for mvtbench

[service.mvt]
viewer = true

[[datasource]]
dbconn = "postgresql://mvtbench:mvtbench@mvtbenchdb/mvtbench"
name = "pg"
default = true

[grid]
predefined = "web_mercator"

[[tileset]]
name = "ne_countries"
attribution = "Natural Earth v4"
extent = [-179.97277, -83.05457, 179.99366, 83.23559]
minzoom = 0
maxzoom = 6

[[tileset.layer]]
name = "country"
geometry_field = "wkb_geometry"
geometry_type = "MULTIPOLYGON"
srid = 3857
buffer_size = 3
simplify = true
[[tileset.layer.query]]
sql = """SELECT wkb_geometry, adm0_a3, mapcolor7 FROM ne_10m_admin_0_countries WHERE min_zoom::integer <= !zoom! AND wkb_geometry &
```

# MapServer

## Mapfile



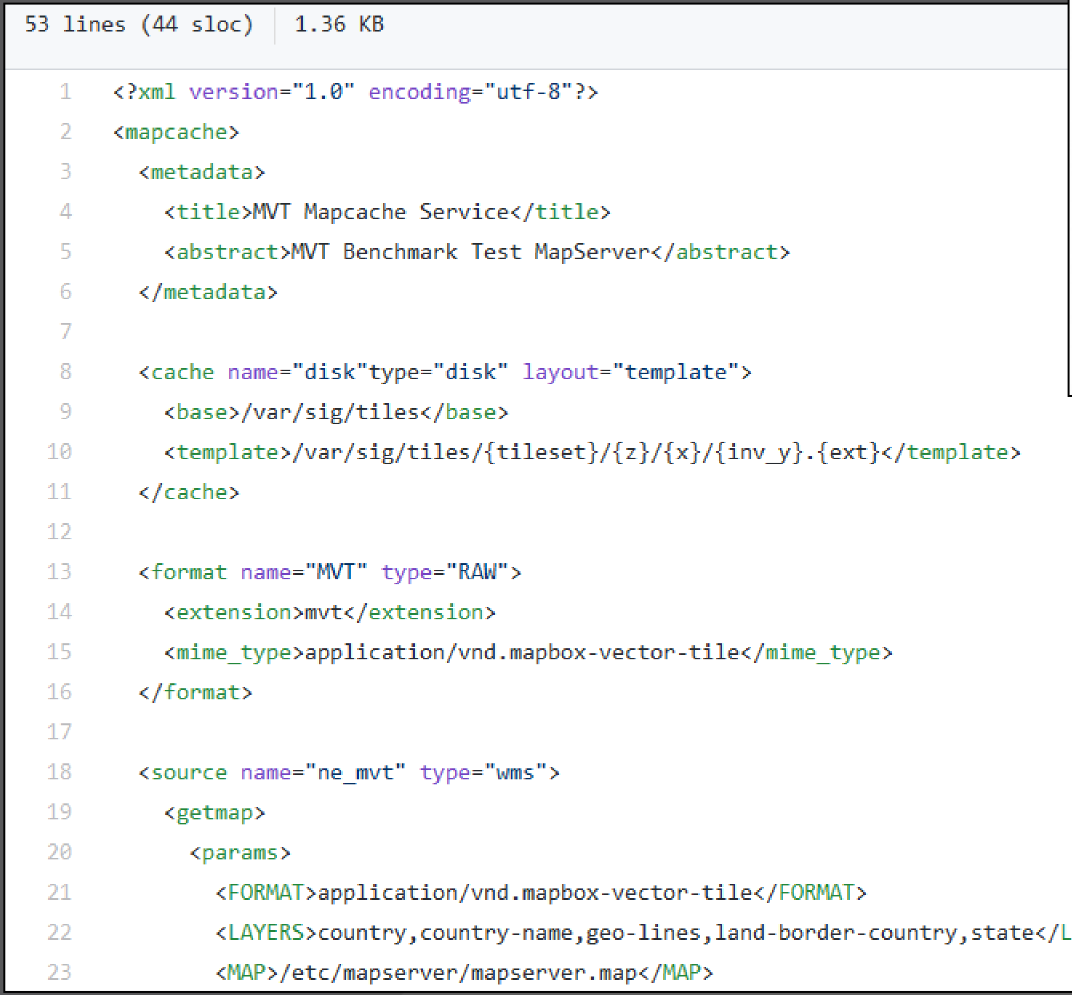
```
OUTPUTFORMAT
  NAME "mvt"
  DRIVER MVT
  FORMATOPTION "EXTENT=512" # default is 4096
  FORMATOPTION "EDGE_BUFFER=20"
END
```



```
LAYER
 NAME "country-name"
 TYPE POINT
 STATUS OFF
 CONNECTIONTYPE postgis
 PROCESSING "CLOSE_CONNECTION=DEFER"
 CONNECTION "user=mvtbench password=mvtbench dbname=mvtbench
 host=mvtbenchdb"
 DATA "wkb_geometry from (SELECT ogc_fid, wkb_geometry, abbrev, name
 FROM ne_10m_admin_0_country_points) as temp using unique ogc_fid using
 SRID=3857"
 EXTENT -19729044.151792 -15878634.348995 19872743.796075
 12257650.087343 # added to improve perfomance
 DUMP true
 METADATA
   "wms_title" "country-name"
   "wms_srs" "epsg:4326 epsg:3857"
   "wms_feature_info_mime_type" "text/html"
   "gml_include_items" "abbrev,name"       # need to be explicit
 END
 PROJECTION
   "init=epsg:3857"
 END
```

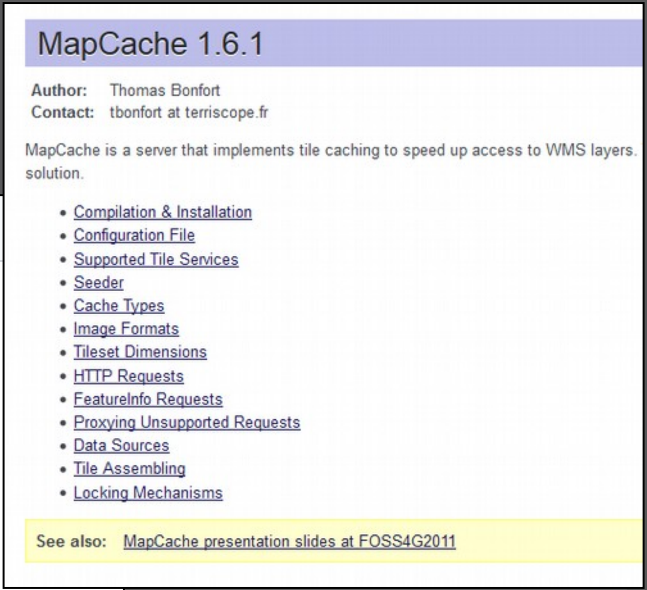
# MapCache

## Mapcache.xml



53 lines (44 sloc) | 1.36 KB

```
<?xml version="1.0" encoding="utf-8"?>
<mapcache>
  <metadata>
    <title>MVT Mapcache Service</title>
    <abstract>MVT Benchmark Test MapServer</abstract>
  </metadata>

  <cache name="disk"type="disk" layout="template">
    <base>/var/sig/tiles</base>
    <template>/var/sig/tiles/{tileset}/{z}/{x}/{inv_y}.{ext}</template>
  </cache>

  <format name="MVT" type="RAW">
    <extension>mvt</extension>
    <mime_type>application/vnd.mapbox-vector-tile</mime_type>
  </format>

  <source name="ne_mvt" type="wms">
    <getmap>
      <params>
        <FORMAT>application/vnd.mapbox-vector-tile</FORMAT>
        <LAYERS>country,country-name,geo-lines,land-border-country,state</LAYERS>
        <MAP>/etc/mapserver/mapserver.map</MAP>
```



### MapCache 1.6.1

**Author:** Thomas Bonfort
**Contact:** tbonfort at terriscope.fr

MapCache is a server that implements tile caching to speed up access to WMS layers. solution.

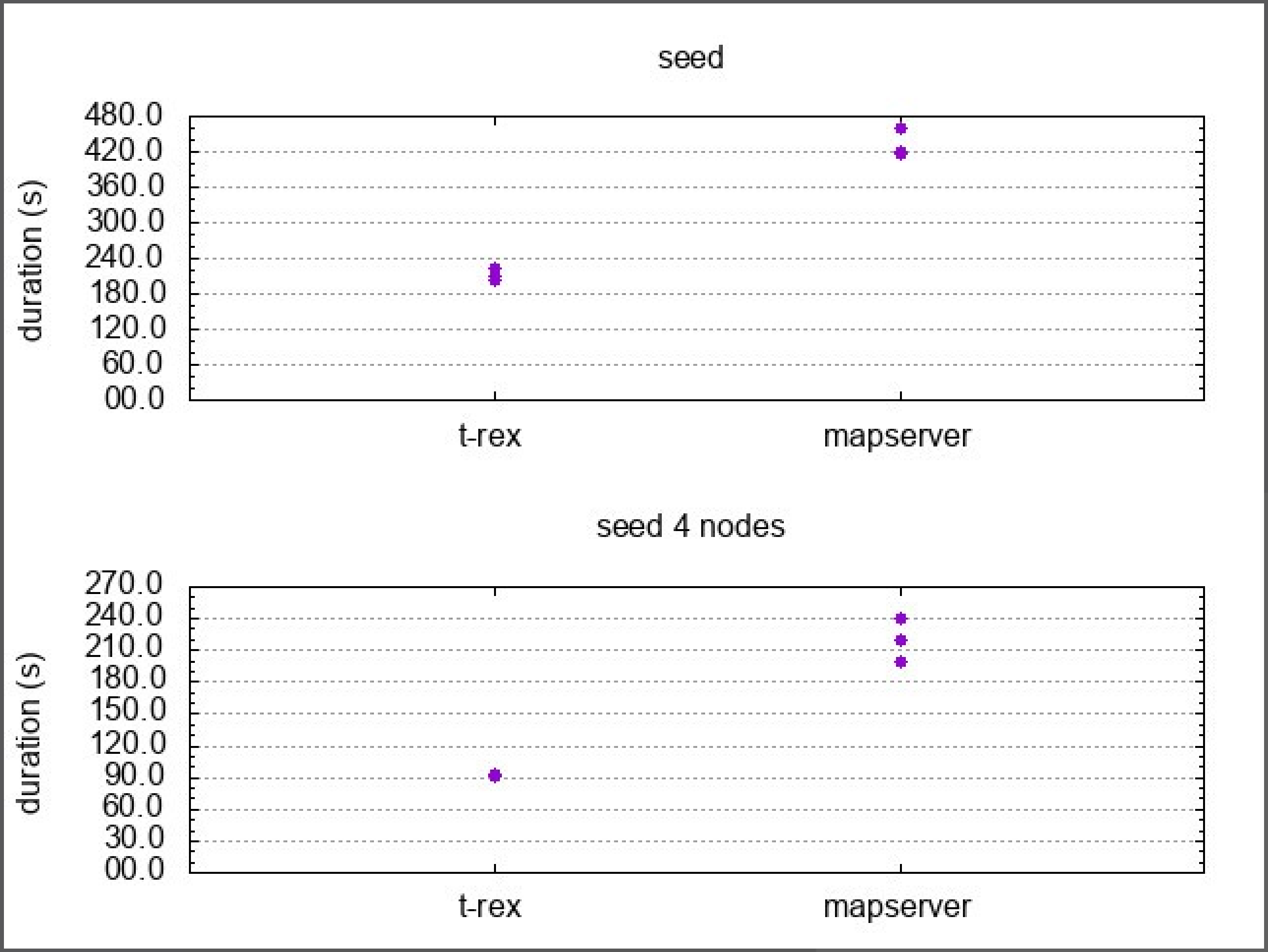
- Compilation & Installation
- Configuration File
- Supported Tile Services
- Seeder
- Cache Types
- Image Formats
- Tileset Dimensions
- HTTP Requests
- FeatureInfo Requests
- Proxying Unsupported Requests
- Data Sources
- Tile Assembling
- Locking Mechanisms

**See also:** MapCache presentation slides at FOSS4G2011

# Test environment

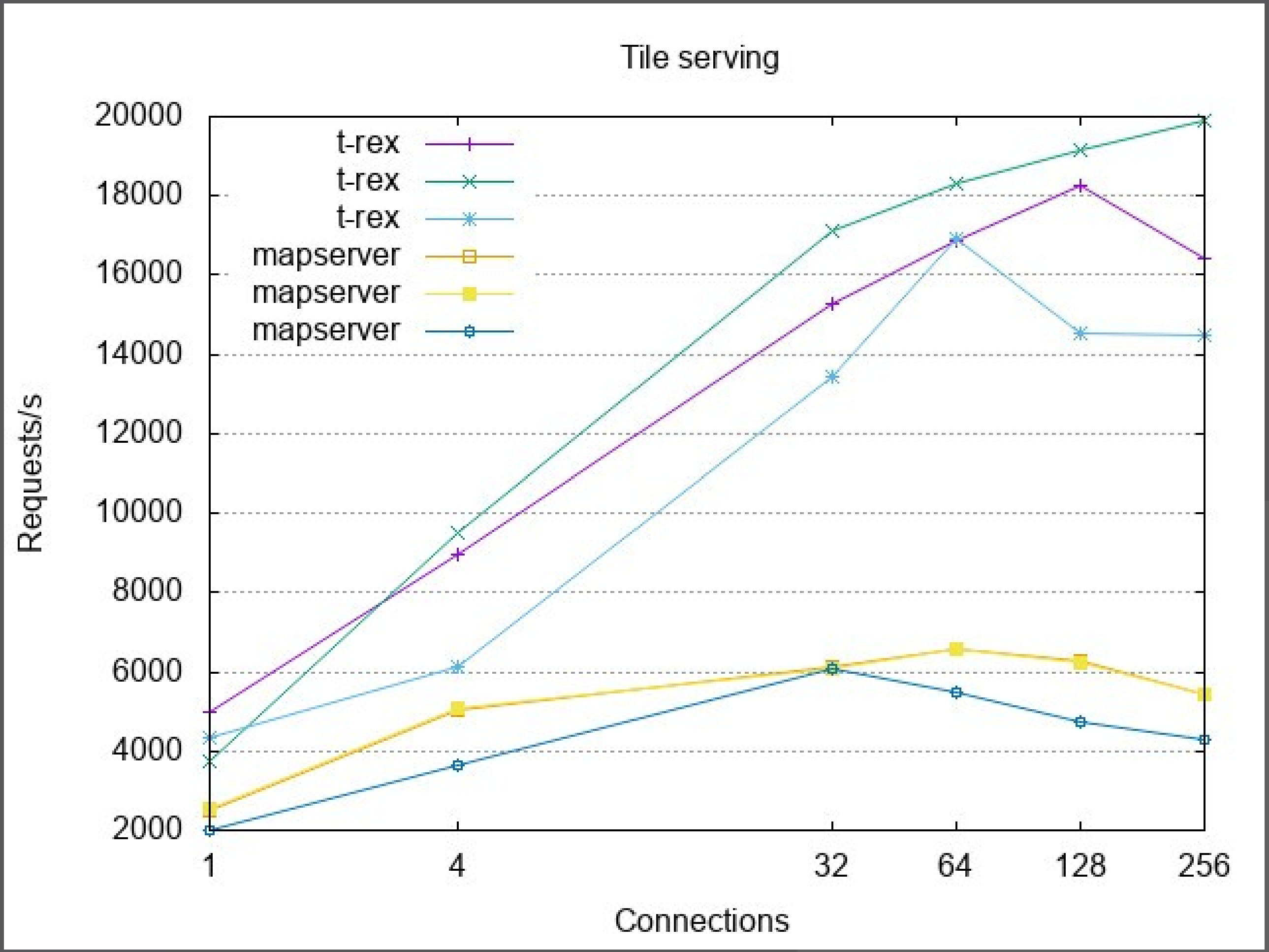
- Linux Notebook with i5-5200U CPU @ 2.20GHz
  2 Cores / 4 Threads
- 8GB RAM
- SSD Disk
- 3 benchmark runs

# Results – Tile seeding



- t-rex: min. 1’32s for 5013 tiles = 45 tiles/s
- MapServer: min. 3’19s for 5461 tiles =  27 t/s

# Results – Tile serving



- t-rex: max. 19’877 tiles/s
- MapServer/Apache: max. 6’585 tiles/s

# Open points

- **Benchmark**
  - Clarify expected number of tiles
  - Adapt scripts for running on multiple servers
  - Measure simplification (visual check only)?
  - Verify clipping?
- **MapServer**
  - Not implemented: data query filter with “z” parameter (use a template?)
  - Ogrinfo error message:

```
Geometry: Multi Point
Feature Count: 1
ERROR 1: Parsing error occurred at line 945
```

# Conclusions

- Benchmarking helps projects to improve
- Benchmarking helps users to see differencate projects
- Open for more contestants
- See you in Calgary!

# Thank you!

Pirmin Kalberer
@implgeo

Karsten Vennemann
Terra GIS LTD
karsten@terragis.net
www.terragis.net