

Kartolog - a smart tile cache

Dražen Odošić
dodobas@geoinfo.geof.hr

Department of Geoinformation Science
Faculty of Geodesy, University of Zagreb

Wien, 15-17.07.2011



SOTM-EU, 2011

- research assistant at Department of Geoinformation Science
- Faculty of Geodesy, University of Zagreb
- email & jabber: `dodobas@geoinfo.geof.hr`
- @dodobas
- open source, web technologies, open data

- we need a fast web map with a lot of layers
- disk space is cheap, let's cache everything
- using Mapnik for spatial data rendering
- some experience with existing tile cache technologies
 - TileCache, MapProxy, GeoWebCache

- wanted to try new “kuulbuzz” technologies
- Q: “Can we use NoSQL database to store tiles and metadata?”
- Q: “What are the benefits of storing more metadata?”

Ingredients:

- 300g Mapnik
- 250g Tornado
- 500g MongoDB
- 2 LTS of Python2 dried tears
- 4 Python2 eggs

Procedure:

- mix in a large bowl
- cook for about 2 weeks



-



- developed by FriendFeed, current version 2.0 (APL2)
- non-blocking asynchronous web server written in Python
- uses epoll/kqueue event notification system
- limited WSGI support (async not supported)



- developed by 10gen, current version 1.8.1 (AGPL3)
- document-oriented storage, BSON format
- database -> collection -> document
- replication, auto-sharding, atomic updates, indexing, ...
- spatial index based on geohash (only points)
- drivers for most major programming languages

- started as an integration of GeoDjango and TileCache
- name: Karta (map) + “-logist” = Kartolog
- in October 2010 rewritten in “kuulbuzz” technologies
- simplify tile management through web interface:
 - layer combining
 - seeding, cache invalidation
 - styling, authorization

Projection defines a tile grid (extent, max. resolution, scale factor)

Project references one projection

Layer part of a project (mapfile, opacity, metasize, metabuffer, ...)

Tile identified by “tilehash”

Tilehash

- MD5 hash of: x,y,z,layers,projection and image format
- indexed in MongoDB

tilehash

status 'blank', 'rendering' or 'done'

layers list of layers

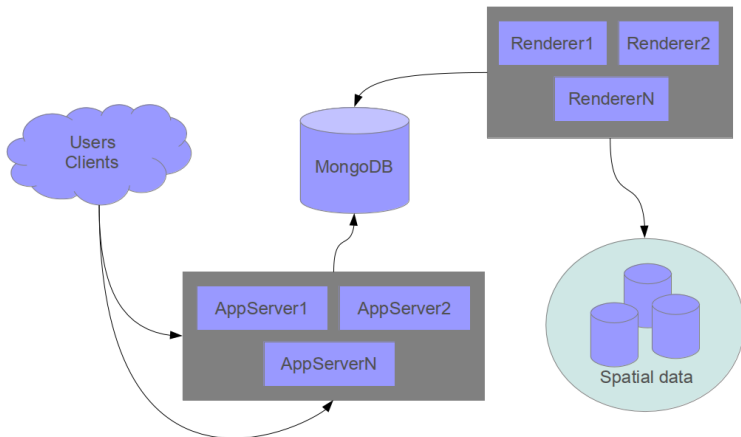
img_format actual image format

accessed number of tile requests

last_access time of last access

- srs, tileSize, x, y, z, when

Kartolog architecture



- WMS “compliant” tile service
- web based management of projections, projects and layers
- simple status report page (number of tiles in db, tiles rendering and blank tiles)
- demo application
- couple of mapreduce scripts for analysis

- support legacy clients, reprojection?
- detect single color images
- more protocols:
 - TMS, WMTS, new Kartolog protocol?
- use Tornado/MongoDB asynchronous features
- prepare statistics scripts and visualization
- better management tools (seeding, invalidating, ...)
- use GridFS?
- real time request logging? (capped collection)
- should we really cache everything?
- documentation ☺

<https://bitbucket.org/dodobas/kartolog>

- used for 8 months
 - 26 layers -> 265 layer combinations
 - 423954 tiles
 - storage: 11.4 Gb (12.9Gb)
 - indexes: 70Mb
 - biggest layer - 28625 tiles, 2.3 Gb, smallest 24, 76Kb