Building SDIs and geoportals with GeoNode and a search engine

FOSS4G 2017, Boston

Paolo Corti, Devika Kakkar, Ben Lewis

Harvard Center for Geographic Analysis
http://gis.harvard.edu/

Authors: Paolo Corti

- works at Harvard CGA
- in the GIS industry since 20 years (10 years with open source)
- OSGeo charter member, GeoNode committer, member of the GeoNode and the pycsw Project Steering Committees
- co-author of the PostGIS Cookbook
- social:
    - Blog: http://www.paolocorti.net
    - Twitter: capooti
    - GitHub: capooti

Please introduce yourself

- Your name and organization
- Your background
- Your experience with open source GIS
- Any experience with web development?
- Any experience with Python development?
- Any experience with SDI, geoportals and OGC standards?
- Any experience with search engines (Solr, Elasticsearch….)

Topics you will explore in this workshop

SDI, Geoportals, OGC web services and standards (WMS, WFS, TMS, CSW, SLD), web mapping, tile caching, spatial databases, search engines, message and task queues

Tools you will use in this workshop

GeoNode, Solr, Python, Django, GeoServer, curl, OWSlib, pycsw, GeoWebCache, PostgreSQL, PostGIS, QGIS, OpenLayers, GDAL/OGR, Celery, RabbitMQ, Vagrant, VirtualBox, git

Workshop Outline 1

Introduction

0. Setup the workshop environment
1. A GeoNode quickstart
2. Programming GeoNode with Python
3. OGC services with GeoServer
4. Tiles caching with GeoWebCache
5. Catalogue services with pycsw

Workshop outline 2

6. Spatial queries with PostGIS
7. Using GeoNode OGC services with external clients
8. Using GeoNode management commands
9. Using a search engine with GeoNode: a quick tour of Solr
10. Developing a custom application for GeoNode
11. Running asynchronous tasks using a task queue (Celery/RabbitMQ)

Where to find workshop material

- GitHub: https://github.com/capooti/foss4g_2017_geonode_solr
- Documentation: http://www.paolocorti.net/foss4g_2017_geonode_solr/

What is required before to start

You should install on your computer (Linux/OSX/Windows):

- git
- Vagrant
- VirtualBox
- QGIS
- Firefox/Chrome

If using Windows you will need to install PuTTy (we did not test the workshop with Windows, though)

Git

Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development

You will use Git to clone the repository of this Workshop and to install GeoNode

VirtualBox

VirtualBox is a cross-platform virtualization application

It extends the capabilities of your existing computer so that it can run multiple operating systems (inside multiple virtual machines) at the same time

For example, you can run Windows and Linux on your Mac, run Windows Server on your Linux server, run Linux on your Windows PC, and so on, all alongside your existing applications. You can install and run as many virtual machines as you like -- the only practical limits are disk space and memory

Vagrant

Vagrant is an open-source software product for building and maintaining portable virtual software development environments, such as VirtualBox, Hyper-V, Docker, VMware, and AWS

You will use Vagrant to create a VirtualBox virtual machine where you will run Ubuntu 16.04 LTS and the open source stack needed for this workshop

QGIS

QGIS is a Desktop GIS application which compares favorably with free and commercial products

QGIS is built on top of and proud to be itself Free and Open Source Software (FOSS)

You will use QGIS to try GeoNode OGC Web Services from an external client

Firefox/Chrome

You will need a browser to test GeoNode and Solr

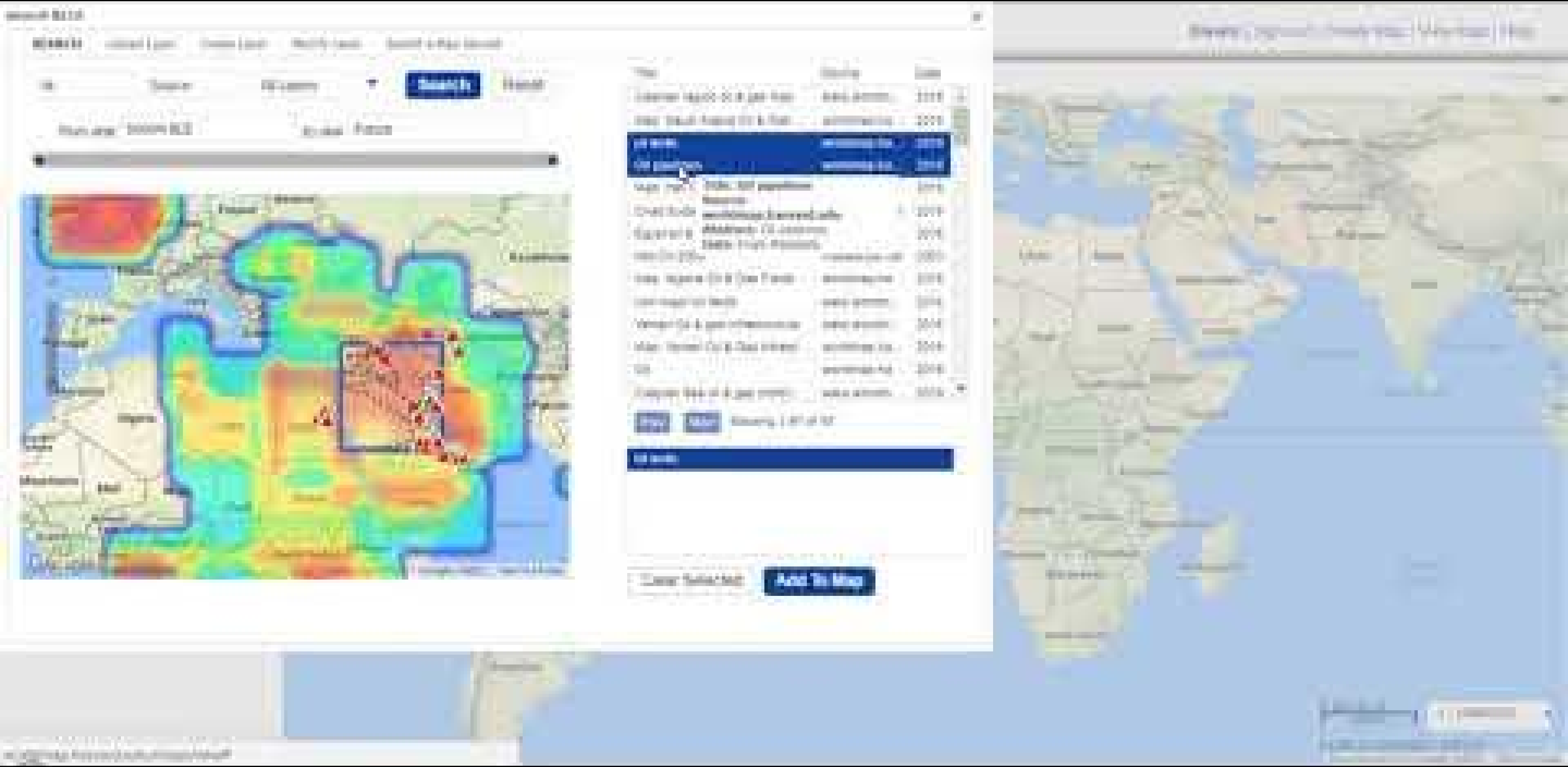
Firefox and Chrome both provide great developer tools which can be used to inspect the request and response cycle between the client and GeoNode

For example the developer tools can be very useful to inspect how the mapping client send a WMS GetMap request to GeoNode

Introduction

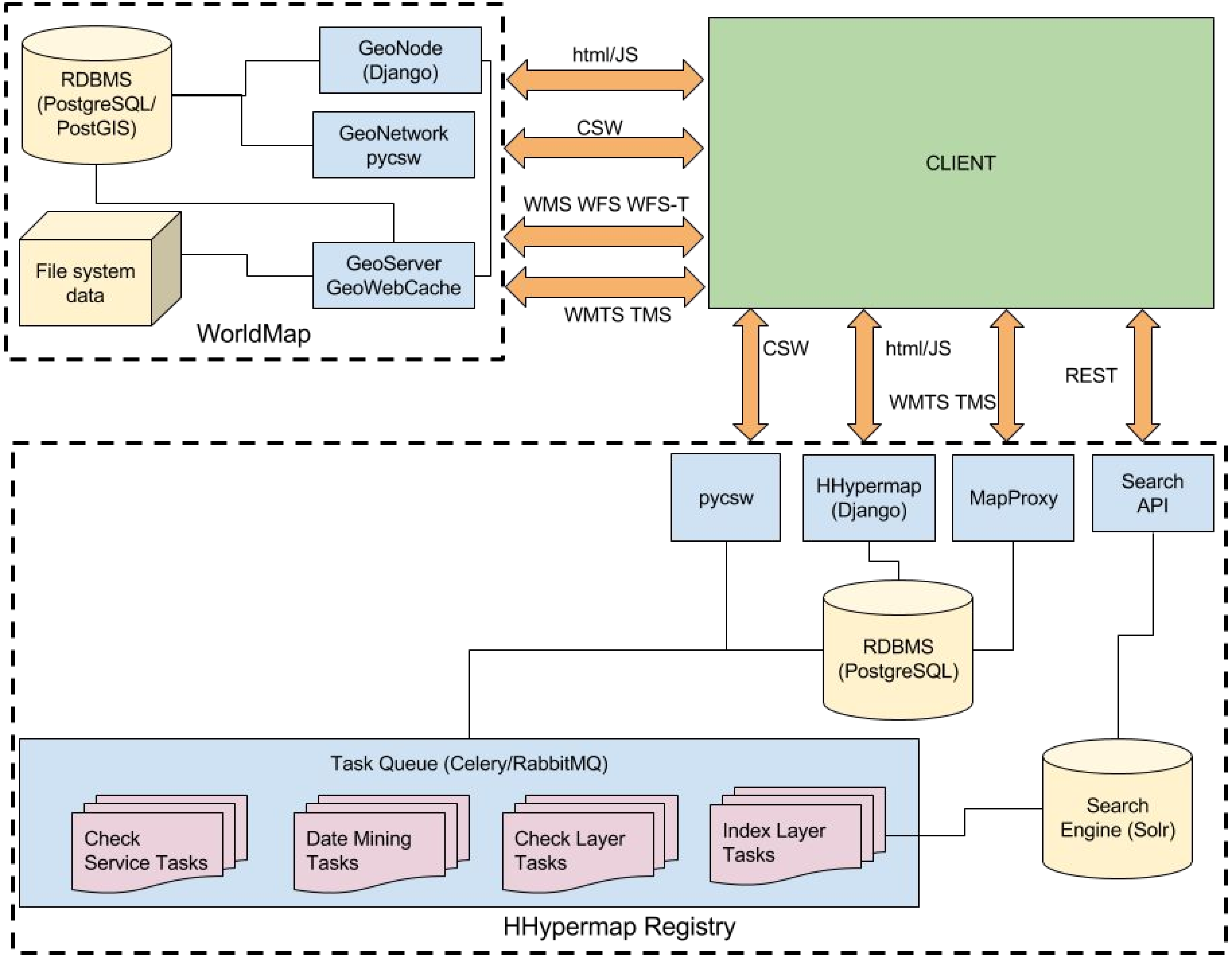
To explain why task queues are important for large SDIs, I will demo WorldMap’s search. With 20,000 local layers and 80,000 remote layers, WorldMap constitutes an SDI that greatly benefits from a scalable back end enabled by task queues.

The back end capabilities which are the focus of this workshop were developed in in service of front end search visualization capabilities designed to allow users to smoothly explore large collections of geodata.


Search
Clear Selected
Add To Map

https://youtu.be/ARtEBEfxOYU

Harvard Hypermap WorldMap Architecture



What is Geonode

GeoNode is a web-based application and platform for developing geospatial information systems (GIS) and for deploying spatial data infrastructures (SDI).

It is designed to be extended and modified, and can be integrated into existing platforms.

GeoNode components

- Django (Python web framework)
- GeoServer, GeoWebCache
- PostgreSQL/PostGIS
- pycsw/GeoNetwork
- GeoExplorer
- Solr/ElasticSearch (optional)
- GeoGig (otpional)

What is Solr

Solr is highly reliable, scalable and fault tolerant, providing distributed indexing, replication and load-balanced querying, automated failover and recovery, centralized configuration and more. Solr powers the search and navigation features of many of the world's largest internet sites. It is built on Lucene.

Solr features

- Full text search, natural language processing, weighted results, fuzzy tolerance results, hit highlighting
- Query suggestions and spelling
- Faceting
- Rich document parsing
- Spatial search
- Schema or schemaless
- Highly scalable and replicable architecture

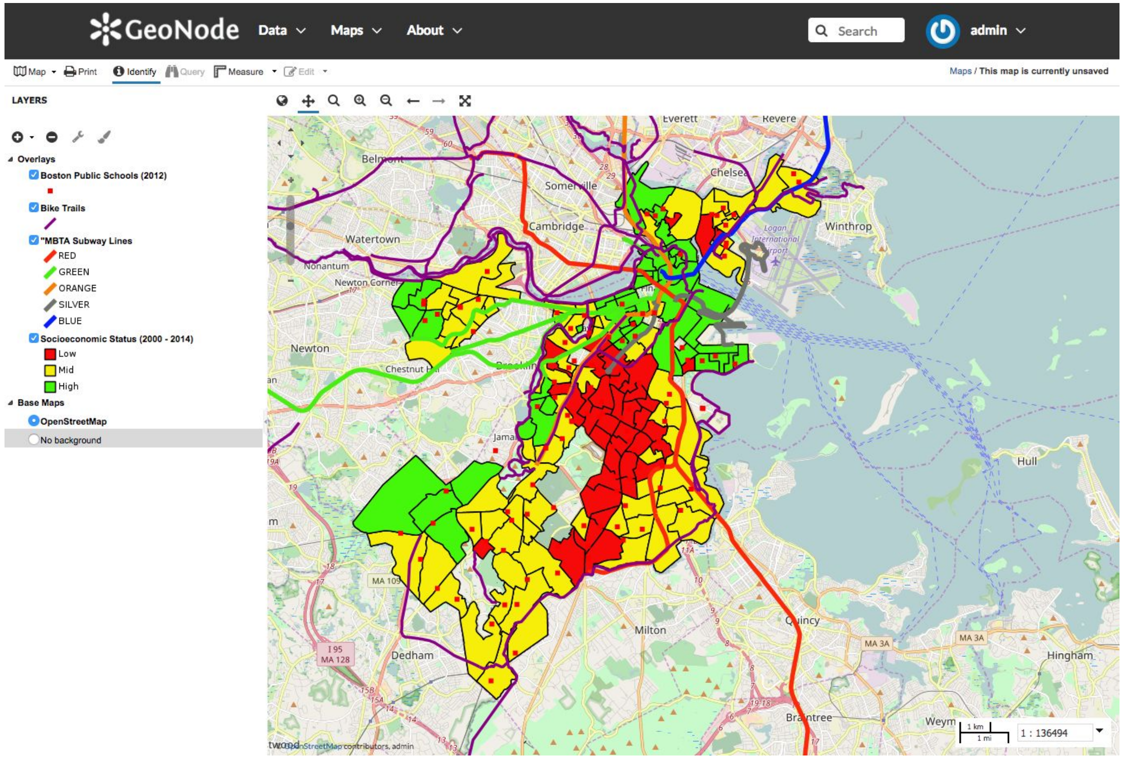
0. Setup the workshop environment

- Git clone the workshop repository
- Start the Vagrant box and SSH into it
- GeoNode installation in a virtual environment
- Solr installation
- PostgreSQL and PostGIS installation

1. GeoNode quickstart

- Overview of the datasets
- Upload datasets to GeoNode
- Metadata edit
- Create a map
- Edit the style of the layers
- Save the map

1. GeoNode quickstart



2. Programming GeoNode with Python

- Start the Django shell
- Play with the Django shell
  - Access layer properties
  - Access map properties
  - Edit metadata
- Access features properties using Python and GDAL/OGR

What you will learn

- How to interact programmatically with Django/GeoNode using the Django shell
- Use Python and Django to read layers and maps properties
- Use Python and Django to edit metadata
- Read features properties and geometries using Python and GDAL/OGR

3. OGC Services with GeoServer

- OGC Web Services
- An overview of the GeoServer web administration interface
- A gentle introduction to the WMS and WFS OGC standards
- The GeoServer REST API

What you will learn

- OGC standards in action with GeoServer
- Using Firefox/Chrome developer toolbar to understand GeoNode requests to GeoServer OGC web services
- Use the OWSLib Python library to interact with OGC web services
- Have a look at the GeoServer REST API and use it using curl and gsconfig

4. Tiles caching with GeoWebCache

- Need for caching
- GeoWebCache GeoServer interface
- GeoWebCache directory
- Requests to the GWC endpoints (WMS-C, TMS, WMTS)
- GeoWebCache REST API

What you will learn

- Why we need to cache WMS
- How tiles caching is implemented in GeoNode
- GWC interface
- How the cache is written to the disk
- How to query the GWC endpoints (WMS-C, TMS, WMTS)
- When you can use the GWC REST API

5. Catalogue services with pycsw

- Using pycsw in GeoNode
- pycsw operations
- Accessing the pycsw catalogue from Python
- Accessing the pycsw catalogue from QGIS

What you will learn

- An introduction to the OGC Catalogue Service for the Web standard (CSW)
- How pycsw is used by GeoNode
- pycsw operations
- Using OWSlib to query GeoNode metadata using pycsw
- Using QGIS MetaSearch to query GeoNode metadata using the pycsw endpoint

6. Spatial query with PostGIS

- Database creation
- Configure GeoNode to use PostgreSQL/PostGIS to store uploaded vector layers
- A quick tour of PostGIS features
  - Querying geometries
  - Transforming data
  - Editing geometries
  - Spatial Joins

What you will learn

- How to create a PostgreSQL database
- How to enable PostGIS in the database
- How to use the PostGIS database as the GeoNode data store
- A quick overview of PostGIS spatial SQL

7. Using GeoNode OGC services with external clients

- QGIS WMS
- QGIS WFS
- OpenLayers
- GDAL/OGR

What you will learn

- How to load a GeoNode WMS in QGIS
- How to load a GeoNode WFS in QGIS
- Create a basic web viewer with OpenLayer displaying a GeoNode layer
- Use GDAL/OGR to query WFS layers and features, and doing features extraction

8. Using GeoNode management commands

- updatelayers
- importlayers

What you will learn

- An overview of GeoNode management commands
- How to run GeoNode management commands
- How to create a spatial view in PostGIS
- How to create a GeoServer layer with a spatial view
- How to register the layer in GeoNode with **updatelayers**
- How to import a shapefile and set some basic metadata with **importlayers**

9. Using a search engine with GeoNode: a quick tour of Solr

- Create a core
- Create a schema
- Indexing data
- Indexing more data
- A quick tour of Solr administrative interface
- Querying Solr

What you will learn

- How to create a core in Solr
- How to create a Solr schema using Python and the Solr schema REST API
- How to index GeoNode data in Solr using Python
- A quick overview of the Solr administrative interface
- How to query Solr (keyword matching, scored results, proximity matching, boosts, date ranges, spatial search, faceting)

10. Developing a custom application for GeoNode

- Create the Django application
- Create a template
- Create the url dispatcher
- Add the application in INSTALLED_APPS
- Create the view
- Run the application

What you will learn

- How to create a Django application and use it in GeoNode
- A basic understanding of Django templates, URL dispatchers and views
- How to query Solr using Python Requests
- How to process the Solr response in the view and render it in the Django template

11. Running asynchronous tasks using a task queue (Celery/RabbitMQ)

- Task queues
- Asynchronous processing model
- Celery and RabbitMQ
- Using Celery in your application
- Using signals
- Process asynchronously with Celery
- Celery monitoring

What you will learn

- What is a asynchronous processing, a task queue, a producer, a consumer and a broker
- Intercepting instance events with Django signals
- How to use Celery and RabbitMQ in GeoNode and in your Django application
- How to monitor Celery tasks using Flower