OSM: From Database to Pictures

Tim Clark (eclipse)

March 12, 2009

What is OSM

What is OSM

- An open source and open data mapping project.

What is OSM

- An open source and open data mapping project.
- Users submit locally collected data and upload it to a central database.

What is OSM

- An open source and open data mapping project.
- Users submit locally collected data and upload it to a central database.
- The database is then used by in various ways by different people.

What is OSM

- An open source and open data mapping project.
- Users submit locally collected data and upload it to a central database.
- The database is then used by in various ways by different people.
- One of these ways is a ‘slippy map’ on their home page.

What is OSM

- An open source and open data mapping project.
- Users submit locally collected data and upload it to a central database.
- The database is then used by in various ways by different people.
- One of these ways is a ‘slippy map’ on their home page.
- For more information on how to contribute to OSM see one rollercow’s past talks.

What is the ‘slippy map’

What is the ‘slippy map’

- Its the map on www.openstreetmap.org.

What is the ‘slippy map’

- Its the map on www.openstreetmap.org.
- Its like most online maps.

What is the ‘slippy map’

- Its the map on www.openstreetmap.org.
- Its like most online maps.
- Most of you are probably familiar with google maps

What is the ‘slippy map’

- Its the map on www.openstreetmap.org.
- Its like most online maps.
- Most of you are probably familiar with google maps
- It allows you to:

What is the ‘slippy map’

- Its the map on www.openstreetmap.org.
- Its like most online maps.
- Most of you are probably familiar with google maps
- It allows you to:
  - You can zoom in.

What is the 'slippy map'

- Its the map on www.openstreetmap.org.
- Its like most online maps.
- Most of you are probably familiar with google maps
- It allows you to:
  - You can zoom in.
  - and out.

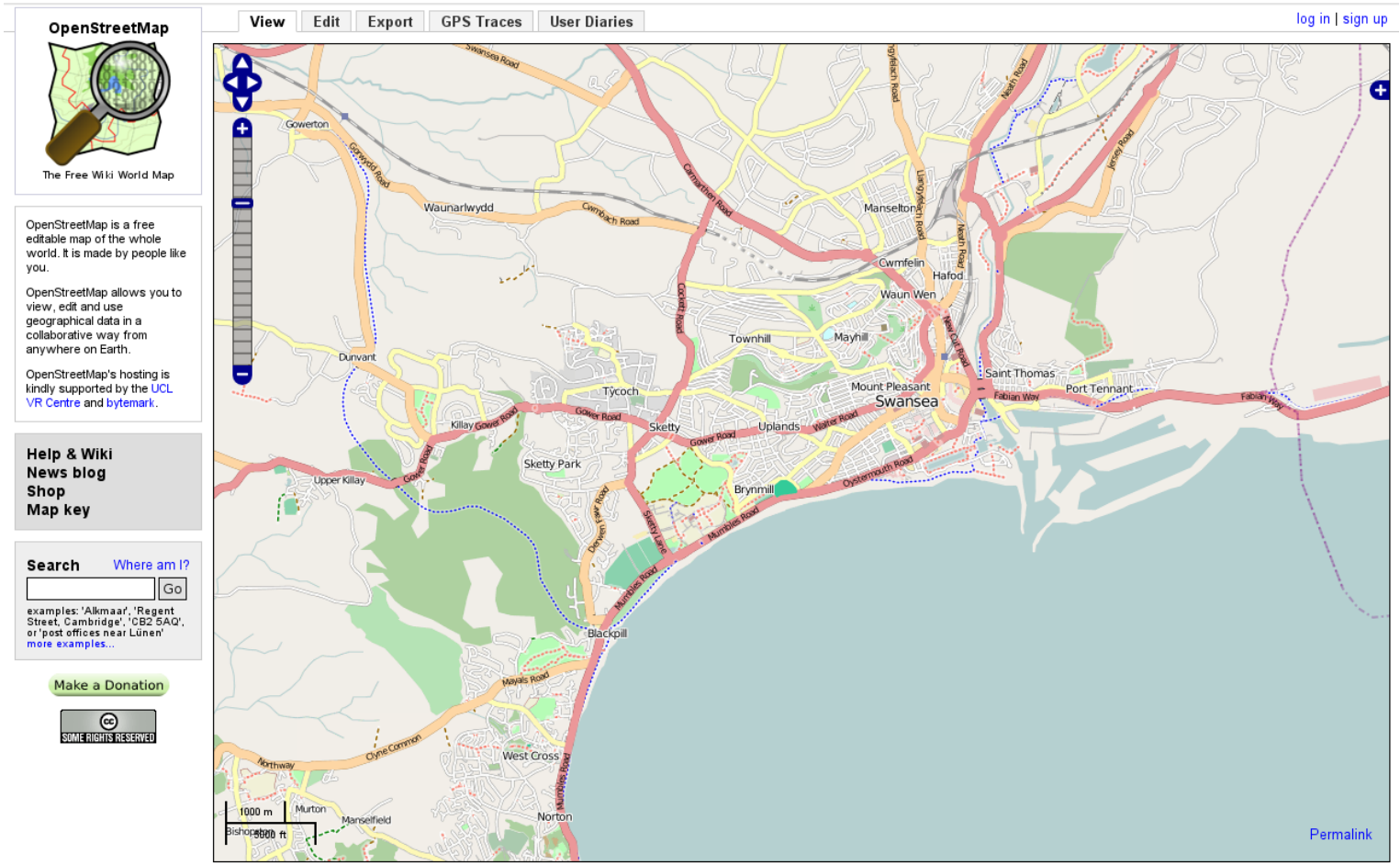
What is the 'slippy map'

- Its the map on www.openstreetmap.org.
- Its like most online maps.
- Most of you are probably familiar with google maps
- It allows you to:
  - You can zoom in.
  - and out.
  - you can drag it round with your mouse.

What is the 'slippy map'

- Its the map on www.openstreetmap.org.
- Its like most online maps.
- Most of you are probably familiar with google maps
- It allows you to:
  - You can zoom in.
  - and out.
  - you can drag it round with your mouse.
- and it looks like this:

OSM Homepage



from www.openstreetmap.org

How does it work

How does it work

- It uses OpenLayers.

How does it work

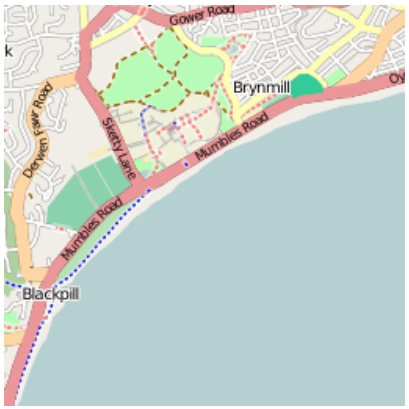
- It uses OpenLayers.
- You point OpenLayers at a tile server that holds (or just serves) lots of tiles.

How does it work

- It uses OpenLayers.
- You point OpenLayers at a tile server that holds (or just serves) lots of tiles.
- Tiles are small sections of map usually 256x256 pixels at a specific zoom level.

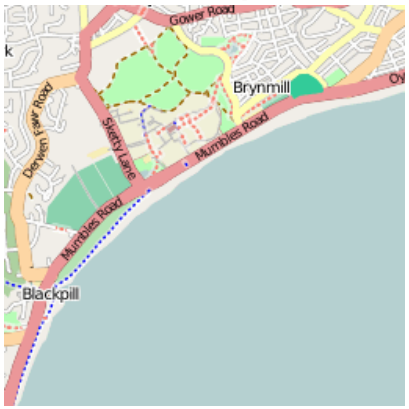
How does it work

- It uses OpenLayers.
- You point OpenLayers at a tile server that holds (or just serves) lots of tiles.
- Tiles are small sections of map usually 256x256 pixels at a specific zoom level.
- For example zoom level 13 tile with campus on it:



How does it work

- It uses OpenLayers.
- You point OpenLayers at a tile server that holds (or just serves) lots of tiles.
- Tiles are small sections of map usually 256x256 pixels at a specific zoom level.
- For example zoom level 13 tile with campus on it:



- This tile is found at tile.openstreetmap.org/13/4005/2720.png

Ways of Rendering

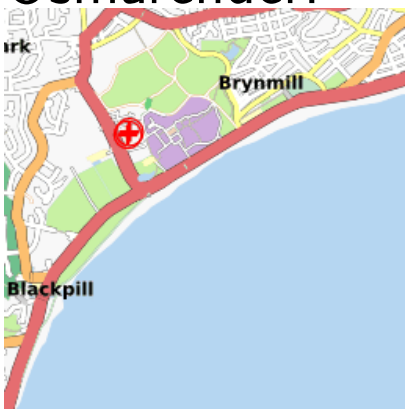
Ways of Rendering

- Mapnik:

Ways of Rendering

- Mapnik:

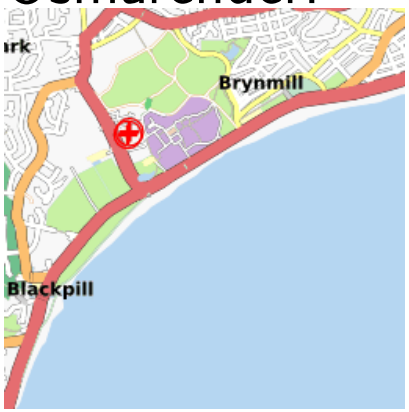
- Osmarender:



Ways of Rendering

- Mapnik:

- Osmarender:



- Mapnik tends to be more popular.

Mapnik

Mapnik

- C Program.

Mapnik

- C Program.
- Therefor very fast.

Mapnik

- C Program.
- Therefor very fast.
- Not just used by OSM.

Mapnik

- C Program.
- Therefor very fast.
- Not just used by OSM.
- Can do OS Maps with it too if you can get the data.

Mapnik

- C Program.
- Therefor very fast.
- Not just used by OSM.
- Can do OS Maps with it too if you can get the data.
- Has python bindings.

Mapnik

- C Program.
- Therefor very fast.
- Not just used by OSM.
- Can do OS Maps with it too if you can get the data.
- Has python bindings.
- Renders out of an GIS database.

Mapnik

- C Program.
- Therefor very fast.
- Not just used by OSM.
- Can do OS Maps with it too if you can get the data.
- Has python bindings.
- Renders out of an GIS database.
- Usually a PostgreSQL database.

Mapnik

- C Program.
- Therefor very fast.
- Not just used by OSM.
- Can do OS Maps with it too if you can get the data.
- Has python bindings.
- Renders out of an GIS database.
- Usually a PostgreSQL database.
- Uses an xml style file which descibes what each type of element looks like.

Mapnik

- C Program.
- Therefor very fast.
- Not just used by OSM.
- Can do OS Maps with it too if you can get the data.
- Has python bindings.
- Renders out of an GIS database.
- Usually a PostgreSQL database.
- Uses an xml style file which descibes what each type of element looks like.
- Renders after each planet file release (more on this later).

Mapnik

- C Program.
- Therefor very fast.
- Not just used by OSM.
- Can do OS Maps with it too if you can get the data.
- Has python bindings.
- Renders out of an GIS database.
- Usually a PostgreSQL database.
- Uses an xml style file which descibes what each type of element looks like.
- Renders after each planet file release (more on this later).
- More about how to get this working later.

Osmarender

- Uses xml transforms on the planet.osm file (more about this file later).

Osmarender

- Uses xml transforms on the planet.osm file (more about this file later).
- Produces an svg that is then cut up and converted into png tiles.

Osmarender

- Uses xml transforms on the planet.osm file (more about this file later).
- Produces an svg that is then cut up and converted into png tiles.
- Very slow to run but generates tiles in batches.

Osmarender

- Uses xml transforms on the planet.osm file (more about this file later).
- Produces an svg that is then cut up and converted into png tiles.
- Very slow to run but generates tiles in batches.
- Used by tiles@home.

Osmarender

- Uses xml transforms on the planet.osm file (more about this file later).
- Produces an svg that is then cut up and converted into png tiles.
- Very slow to run but generates tiles in batches.
- Used by tiles@home.
- tiles@home is a cloud render farm for the osmarender tiles on the main map.

Osmarender

- Uses xml transforms on the planet.osm file (more about this file later).
- Produces an svg that is then cut up and converted into png tiles.
- Very slow to run but generates tiles in batches.
- Used by tiles@home.
- tiles@home is a cloud render farm for the osmarender tiles on the main map.
- Rendered continuously as tiles get expired.

Osmarender

- Uses xml transforms on the planet.osm file (more about this file later).
- Produces an svg that is then cut up and converted into png tiles.
- Very slow to run but generates tiles in batches.
- Used by tiles@home.
- tiles@home is a cloud render farm for the osmarender tiles on the main map.
- Rendered continuously as tiles get expired.

Planet.osm

Planet.osm

- XML file.

Planet.osm

- XML file.
- Holds all the current OSM data.

Planet.osm

- XML file.
- Holds all the current OSM data.
- Created every Wednesday.

Planet.osm

- XML file.
- Holds all the current OSM data.
- Created every Wednesday.
- Takes several hours to create and bzip.

Planet.osm

- XML file.
- Holds all the current OSM data.
- Created every Wednesday.
- Takes several hours to create and bzip.
- Its a very big file, about 150 gigabytes

Planet.osm

- XML file.
- Holds all the current OSM data.
- Created every Wednesday.
- Takes several hours to create and bzip.
- Its a very big file, about 150 gigabytes
- Compressed to 5.2GB with bzip2 compression.

Planet.osm

- XML file.
- Holds all the current OSM data.
- Created every Wednesday.
- Takes several hours to create and bzip.
- Its a very big file, about 150 gigabytes
- Compressed to 5.2GB with bzip2 compression.
- Current and historical planet.osm files stored at planet.openstreetmap.org

Deltas

Deltas

- Created with Osmosis.

Deltas

- Created with Osmosis.
- Created on different tile intervals from the history data in the main database:

Deltas

- Created with Osmosis.
- Created on different tile intervals from the history data in the main database:
  - Daily: 15MB-25MB

Deltas

- Created with Osmosis.
- Created on different tile intervals from the history data in the main database:
  - Daily: 15MB-25MB
  - Hourly: 300KB-2MB

Deltas

- Created with Osmosis.
- Created on different tile intervals from the history data in the main database:
  - Daily: 15MB-25MB
  - Hourly: 300KB-2MB
  - Minutely: 5KB - 50KB

Deltas

- Created with Osmosis.
- Created on different tile intervals from the history data in the main database:
  - Daily: 15MB-25MB
  - Hourly: 300KB-2MB
  - Minutely: 5KB - 50KB
- Osmosis can be used to construct an up to date planet.osm file with these.

Deltas

- Created with Osmosis.
- Created on different tile intervals from the history data in the main database:
  - Daily: 15MB-25MB
  - Hourly: 300KB-2MB
  - Minutely: 5KB - 50KB
- Osmosis can be used to construct an up to date planet.osm file with these.
- Can also be used with OSM2PGSQL (more on this next).

OSM2PGSQL

OSM2PGSQL

- Used to put planet.osm files into a GIS PostgreSQL database.

OSM2PGSQL

- Used to put planet.osm files into a GIS PostgreSQL database.
- More recently deltas can be applied to a postgis database with it too.

OSM2PGSQL

- Used to put planet.osm files into a GIS PostgreSQL database.
- More recently deltas can be applied to a postgis database with it too.
- Slim mode:

OSM2PGSQL

- Used to put planet.osm files into a GIS PostgreSQL database.
- More recently deltas can be applied to a postgis database with it too.
- Slim mode:
  - Needed if you want to apply deltas to your database.

OSM2PGSQL

- Used to put planet.osm files into a GIS PostgreSQL database.
- More recently deltas can be applied to a postgis database with it too.
- Slim mode:
  - Needed if you want to apply deltas to your database.
  - Uses much less RAM when importing.

OSM2PGSQL

- Used to put planet.osm files into a GIS PostgreSQL database.
- More recently deltas can be applied to a postgis database with it too.
- Slim mode:
  - Needed if you want to apply deltas to your database.
  - Uses much less RAM when importing.
  - The database ends up being much bigger.

OSM2PGSQL

- Used to put planet.osm files into a GIS PostgreSQL database.
- More recently deltas can be applied to a postgis database with it too.
- Slim mode:
  - Needed if you want to apply deltas to your database.
  - Uses much less RAM when importing.
  - The database ends up being much bigger.
  - It takes much longer to do an import.

OSM2PGSQL

- Used to put planet.osm files into a GIS PostgreSQL database.
- More recently deltas can be applied to a postgis database with it too.
- Slim mode:
  - Needed if you want to apply deltas to your database.
  - Uses much less RAM when importing.
  - The database ends up being much bigger.
  - It takes much longer to do an import.
- Tile deltas:

OSM2PGSQL

- Used to put planet.osm files into a GIS PostgreSQL database.
- More recently deltas can be applied to a postgis database with it too.
- Slim mode:
  - Needed if you want to apply deltas to your database.
  - Uses much less RAM when importing.
  - The database ends up being much bigger.
  - It takes much longer to do an import.
- Tile deltas:
  - New feature.

OSM2PGSQL

- Used to put planet.osm files into a GIS PostgreSQL database.
- More recently deltas can be applied to a postgis database with it too.
- Slim mode:
  - Needed if you want to apply deltas to your database.
  - Uses much less RAM when importing.
  - The database ends up being much bigger.
  - It takes much longer to do an import.
- Tile deltas:
  - New feature.
  - Added by FireFury.

OSM2PGSQL

- Used to put planet.osm files into a GIS PostgreSQL database.
- More recently deltas can be applied to a postgis database with it too.
- Slim mode:
  - Needed if you want to apply deltas to your database.
  - Uses much less RAM when importing.
  - The database ends up being much bigger.
  - It takes much longer to do an import.
- Tile deltas:
  - New feature.
  - Added by FireFury.
  - Produces files that say which tiles are affected by a osm delta.

OSM2PGSQL

- Used to put planet.osm files into a GIS PostgreSQL database.
- More recently deltas can be applied to a postgis database with it too.
- Slim mode:
  - Needed if you want to apply deltas to your database.
  - Uses much less RAM when importing.
  - The database ends up being much bigger.
  - It takes much longer to do an import.
- Tile deltas:
  - New feature.
  - Added by FireFury.
  - Produces files that say which tiles are affected by a osm delta.

Python Shaped Glue

Python Shaped Glue

- updater.py:

Python Shaped Glue

- updater.py:
  - Written by FireFury

Python Shaped Glue

- updater.py:
  - Written by FireFury
  - Modified by me to be a little more sane/useful

Python Shaped Glue

- updater.py:
  - Written by FireFury
  - Modified by me to be a little more sane/useful
  - Downloads osm deltas and calls osm2pgsql on them

Python Shaped Glue

- updater.py:
  - Written by FireFury
  - Modified by me to be a little more sane/useful
  - Downloads osm deltas and calls osm2pgsql on them
  - We have this running on iodine

Python Shaped Glue

- updater.py:
  - Written by FireFury
  - Modified by me to be a little more sane/useful
  - Downloads osm deltas and calls osm2pgsql on them
  - We have this running on iodine
- Cluster render server:

Python Shaped Glue

- updater.py:
  - Written by FireFury
  - Modified by me to be a little more sane/useful
  - Downloads osm deltas and calls osm2pgsql on them
  - We have this running on iodine
- Cluster render server:
  - renderer.py is the render controller: it takes requests to have tiles rendered and passes them out to the workers.

Python Shaped Glue

- updater.py:
  - Written by FireFury
  - Modified by me to be a little more sane/useful
  - Downloads osm deltas and calls osm2pgsql on them
  - We have this running on iodine
- Cluster render server:
  - renderer.py is the render controller: it takes requests to have tiles rendered and passes them out to the workers.
  - worker.py is the render worker: it connects to the render controller and renders the tiles it requests to the NFS by using the mapnik bindings with the postgis database.

Python Shaped Glue

- updater.py:
  - Written by FireFury
  - Modified by me to be a little more sane/useful
  - Downloads osm deltas and calls osm2pgsql on them
  - We have this running on iodine
- Cluster render server:
  - renderer.py is the render controller: it takes requests to have tiles rendered and passes them out to the workers.
  - worker.py is the render worker: it connects to the render controller and renders the tiles it requests to the NFS by using the mapnik bindings with the postgis database.
  - The render controller runs on silver (under nice).

Python Shaped Glue

- updater.py:
  - Written by FireFury
  - Modified by me to be a little more sane/useful
  - Downloads osm deltas and calls osm2pgsql on them
  - We have this running on iodine
- Cluster render server:
  - renderer.py is the render controller: it takes requests to have tiles rendered and passes them out to the workers.
  - worker.py is the render worker: it connects to the render controller and renders the tiles it requests to the NFS by using the mapnik bindings with the postgis database.
  - The render controller runs on silver (under nice).
  - The Workers run on the desktops (under nice) apart from iodine and arsenic.

Python Shaped Glue

- updater.py:
  - Written by FireFury
  - Modified by me to be a little more sane/useful
  - Downloads osm deltas and calls osm2pgsql on them
  - We have this running on iodine
- Cluster render server:
  - renderer.py is the render controller: it takes requests to have tiles rendered and passes them out to the workers.
  - worker.py is the render worker: it connects to the render controller and renders the tiles it requests to the NFS by using the mapnik bindings with the postgis database.
  - The render controller runs on silver (under nice).
  - The Workers run on the desktops (under nice) apart from iodine and arsenic.

Summary

Summary

- OSM release planet file (or planet delta)

Summary

- OSM release planet file (or planet delta)
- updater downloads this and calls OSM2PGSQL

Summary

- OSM release planet file (or planet delta)
- updater downloads this and calls OSM2PGSQL
- which puts the data into the postgis database and makes a tilediff file.

Summary

- OSM release planet file (or planet delta)
- updater downloads this and calls OSM2PGSQL
- which puts the data into the postgis database and makes a tilediff file.
- The nightly job adds all swanseas tiles to the render queue

Summary

- OSM release planet file (or planet delta)
- updater downloads this and calls OSM2PGSQL
- which puts the data into the postgis database and makes a tilediff file.
- The nightly job adds all swanseas tiles to the render queue
- which then spreads the tile requests out to the workers

Summary

- OSM release planet file (or planet delta)
- updater downloads this and calls OSM2PGSQL
- which puts the data into the postgis database and makes a tilediff file.
- The nightly job adds all swanseas tiles to the render queue
- which then spreads the tile requests out to the workers
- which then call mapnik

Summary

- OSM release planet file (or planet delta)
- updater downloads this and calls OSM2PGSQL
- which puts the data into the postgis database and makes a tilediff file.
- The nightly job adds all swanseas tiles to the render queue
- which then spreads the tile requests out to the workers
- which then call mapnik
- which uses the data in the postgis database to render the tile to my home dir.

Summary

- OSM release planet file (or planet delta)
- updater downloads this and calls OSM2PGSQL
- which puts the data into the postgis database and makes a tilediff file.
- The nightly job adds all swanseas tiles to the render queue
- which then spreads the tile requests out to the workers
- which then call mapnik
- which uses the data in the postgis database to render the tile to my home dir.
- A user goes to the website and it open OpenLayers which loads the tiles in a nice UI.

The Internet

Slides Available at `http://sucs.org/~eclipse`
My OSM nightly render: `http://sucs.org/~eclipse/osm`

Questions?

Any Questions?