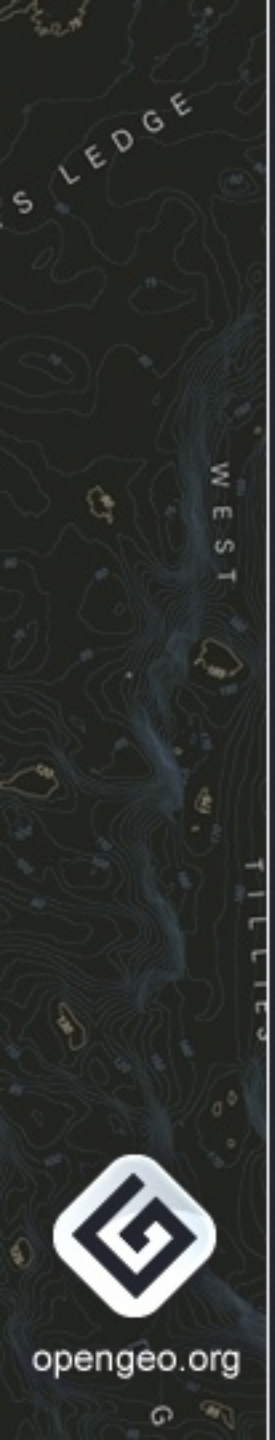
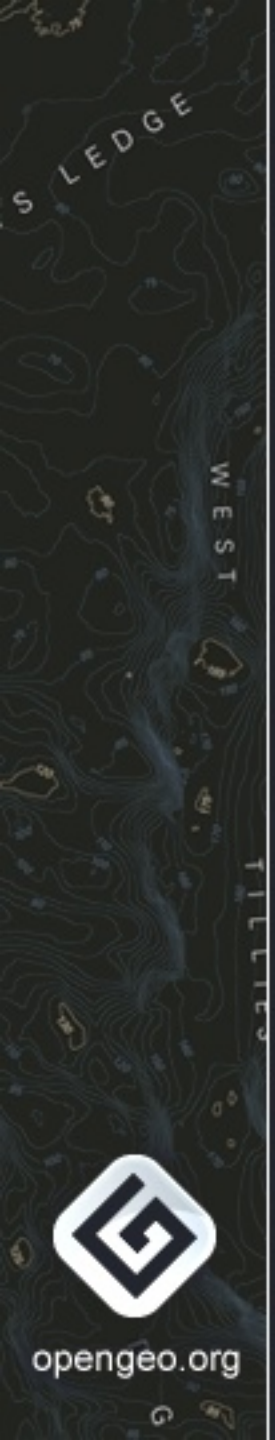


Comparing the Performance of Open Source Web Map Servers

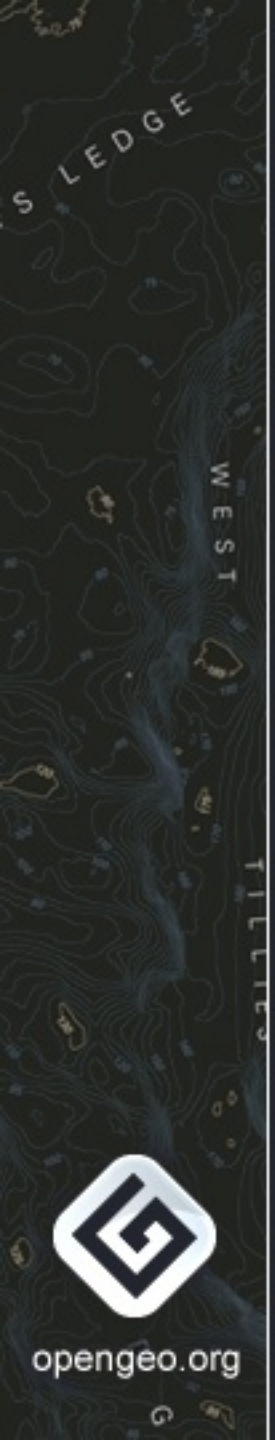
Andrea Aime - OpenGeo

Justin Deoliveira - OpenGeo



Outline

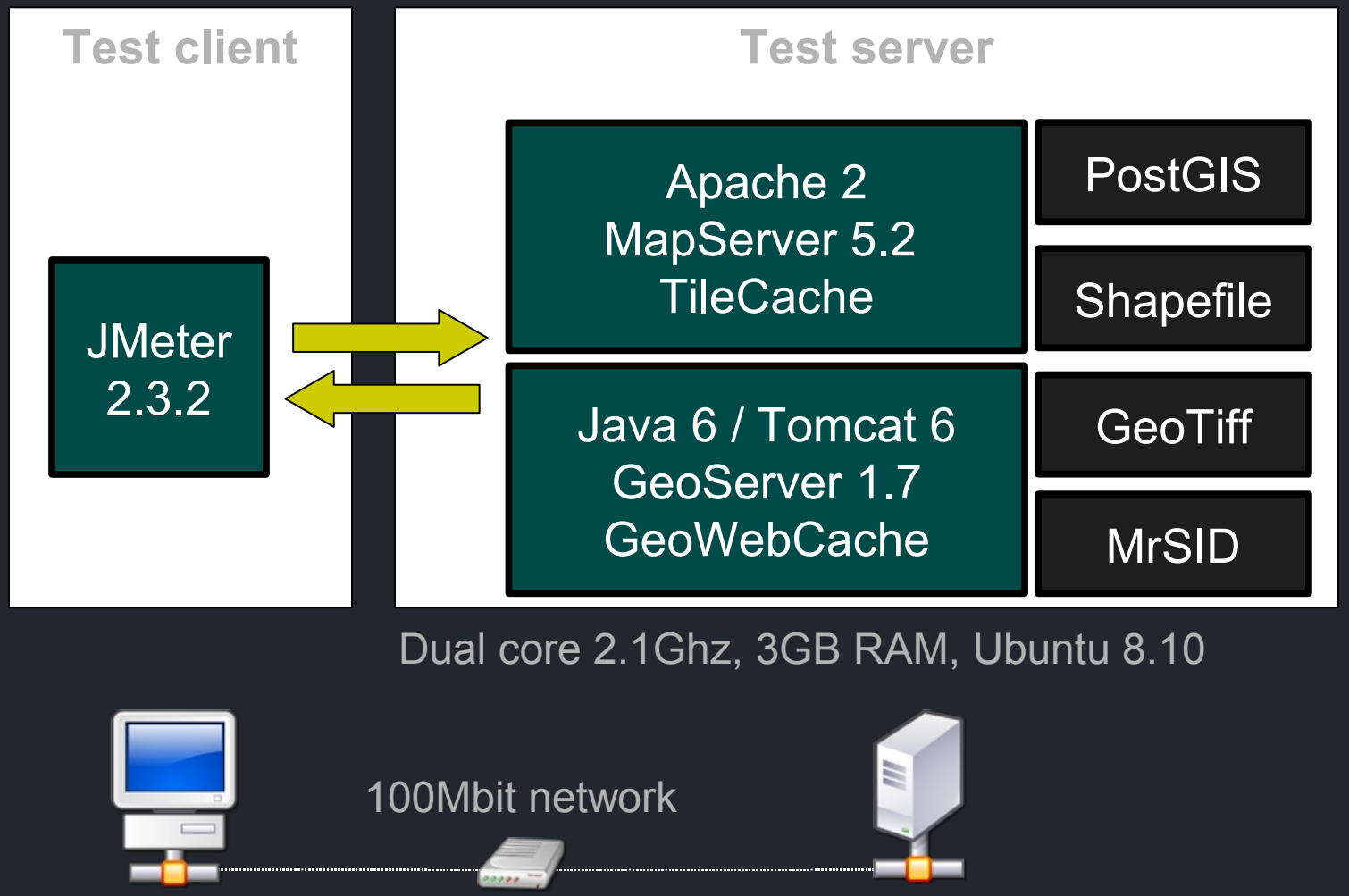
- Test enviroment and setup
- WMS tests
  - vector data (plain and thematic maps)
  - raster data
- Tile cache tests
- WFS tests



Test objectives

- Find out the best performing configurations *(setup matters!)* and making sure the servers are doing the same job
- Assess performance levels on various common tasks not tested at the FOSS4G 2007 performance comparison:
  - thematic maps
  - raster data
  - tile caching
  - WFS
- Provide feedback to the respective developer and user communities

Test enviroment



opengeo.org

Test methodology

- Scale up: 1, 10, 20, 40 client threads
- Tests are repeated multiple times (cold testing does not make sense with Java servers, JIT needs time)
- Make sure the network is not playing the bottleneck (100MBit being a bottleneck? yeah! The theoretical max speed is “only” 12.5MByte/s you know?)
- Compare result quality, not just times
  - this actually uncovered bugs/potential for improvement in both servers

opengeo.org

Setting up MapServer 5.2

- Rebuild from sources:
  - to enable fastcgi*
  - to get MrSID support in GDAL and use the same GDAL version as GeoServer (GDAL 1.4.2)
- Move used EPSG codes at the top of the Proj / usr/share/proj/epsg file (this is crucial to get best performance)
- Use server based fastcgi, 20 mapserver processes always ready to serve requests, each one holding a single database connection (dynamic fastcgi did not work)

**: very important to get good performance against PostGIS, a 3x speedup factor was measured*

opengeo.org

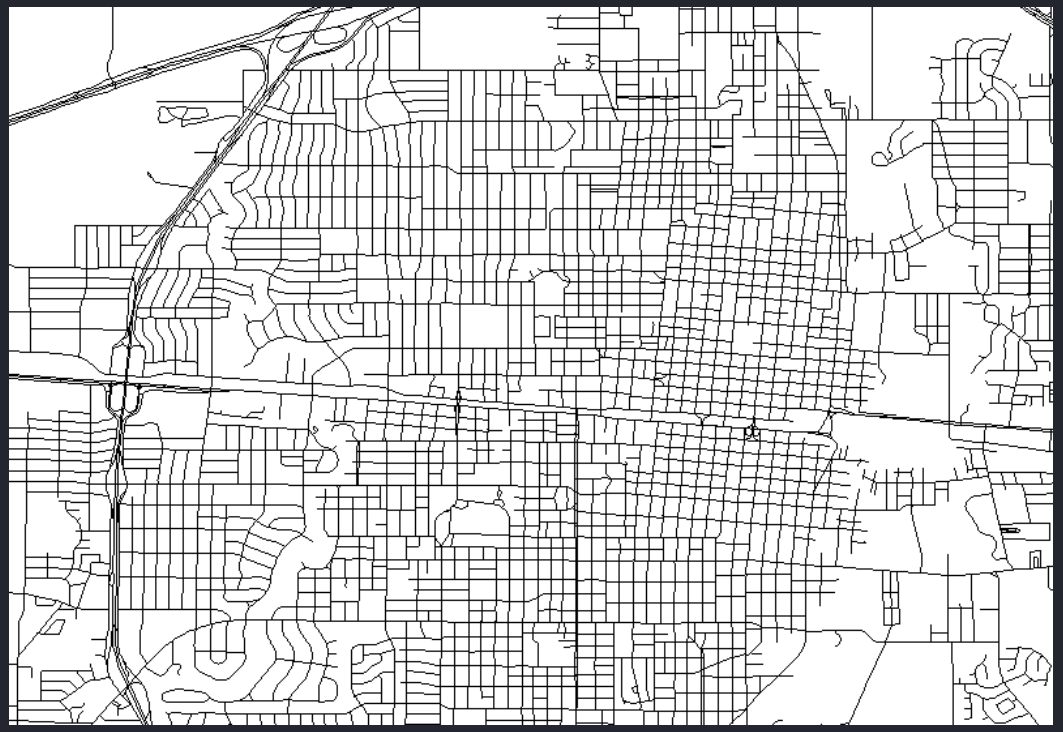
Setting up GeoServer 1.7

- Install Sun Java 6, JAI, JAI Image I/O, and ImageIO-Ext native extensions
- Install latest Tomcat release
- Disable gzip compression filters (useful on a real internet connection, but not in a local network case)
- Make sure exactly 20 threads answers requests, and that exactly 20 dbms connections are used, to match MapServer setup
- No Java VM tweaking needed, the machine is recognized as server class, the VM self tunes in this case (Linux is marked as a server OS, dual core, has more than 2GB RAM)

opengeo.org

Test #1: PostgGIS vs Shapefiles

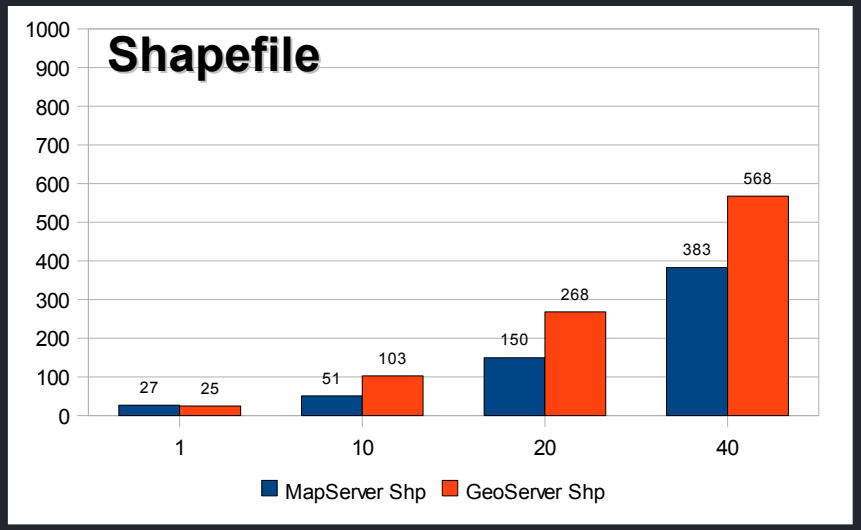
- 3 million roads in Texas
- Each test paints 1000 roads using a small bbox (512x512 tiles), minimal styling, no antialiasing

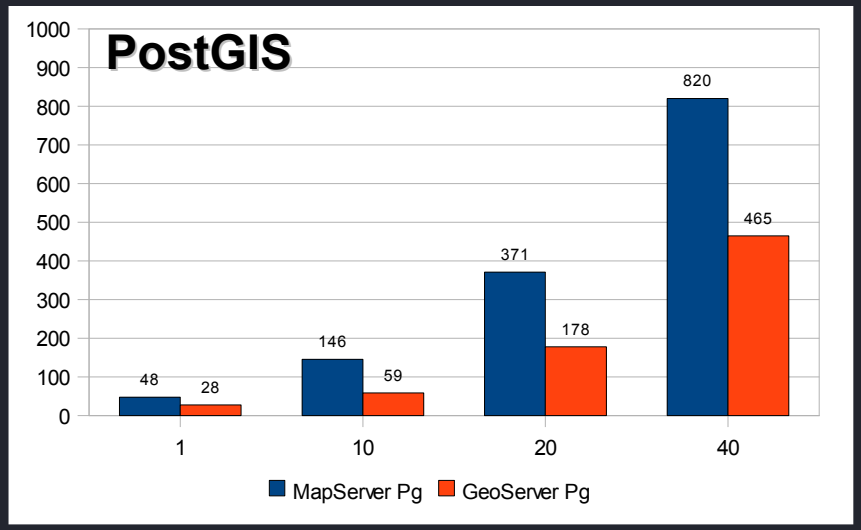


data preview

opengeo.org

#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

opengeo.org

#1: Test output samples

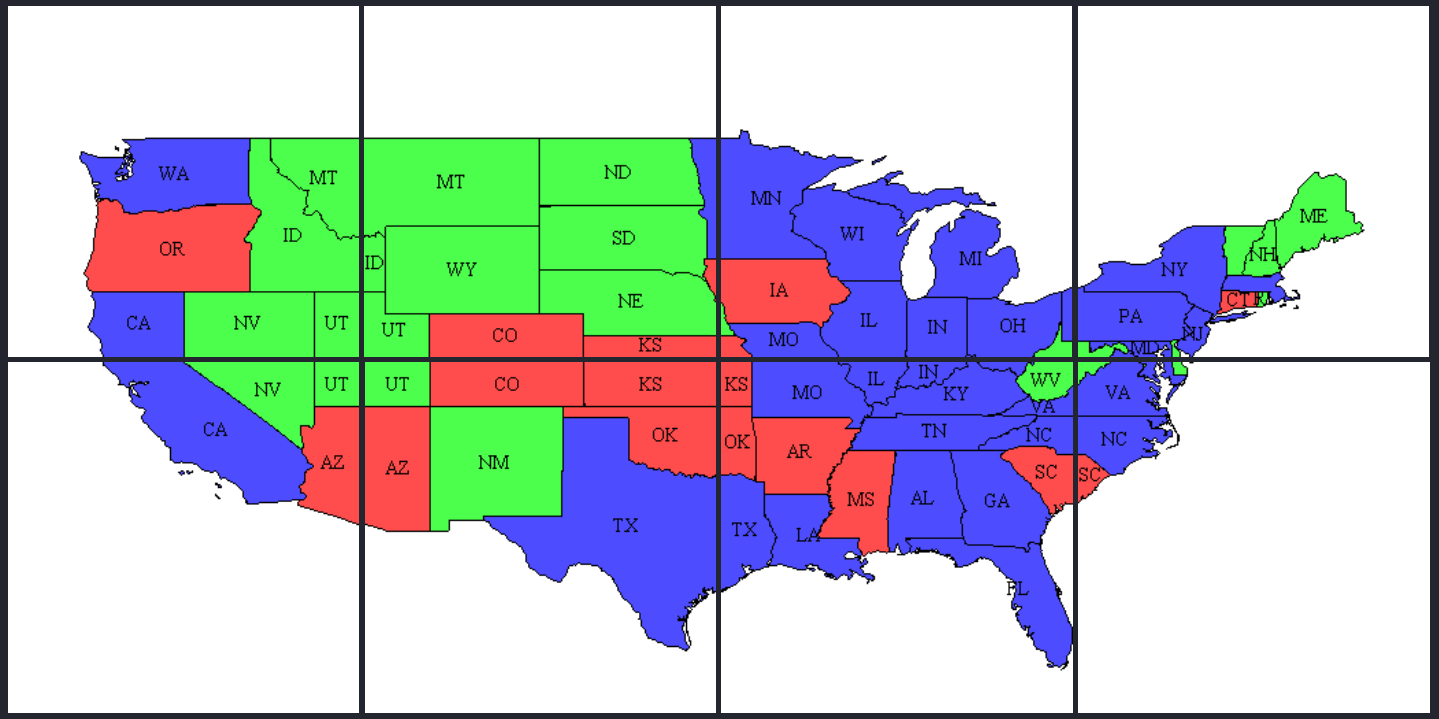
GeoServer (7.8 KB)

MapServer (9 KB)

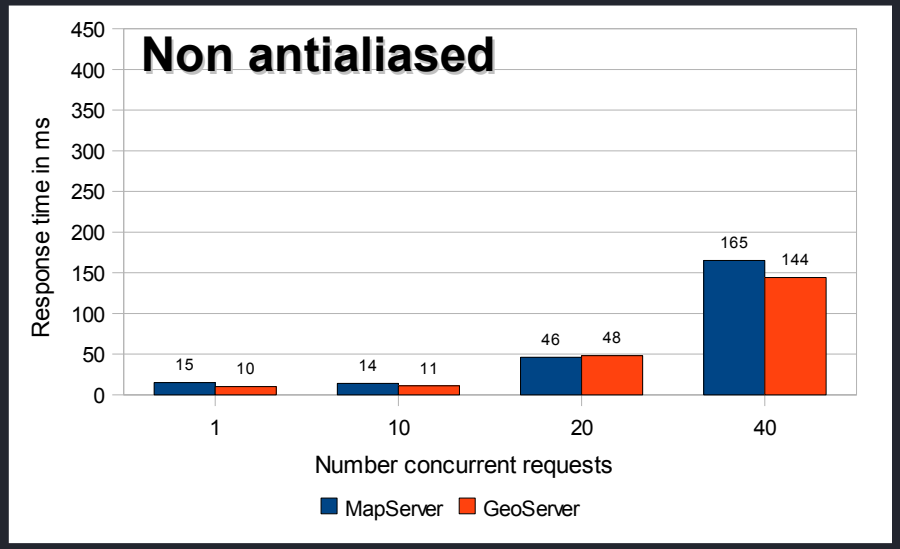
- No significant differences in output quality

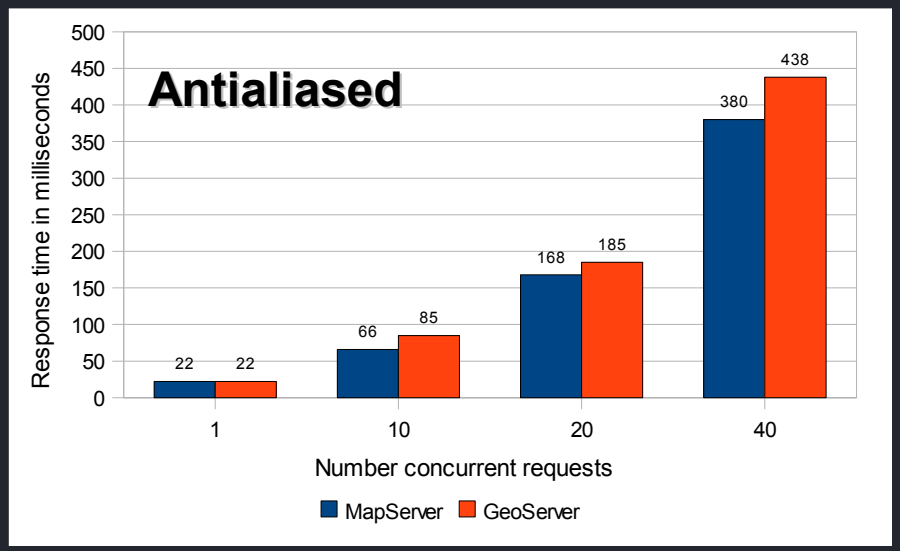
#2: Thematic map

- A USA thematic map, classified by population, with labels
- 8 256x256 tiles, mimickying OpenLayers requests



#2: Performance results

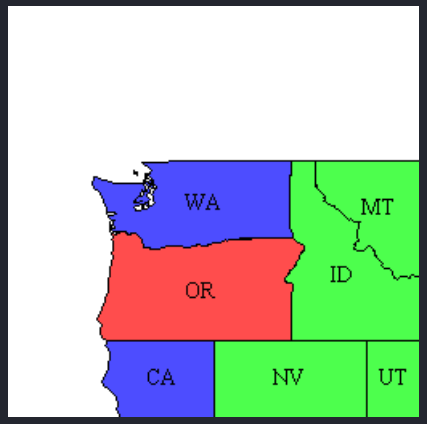




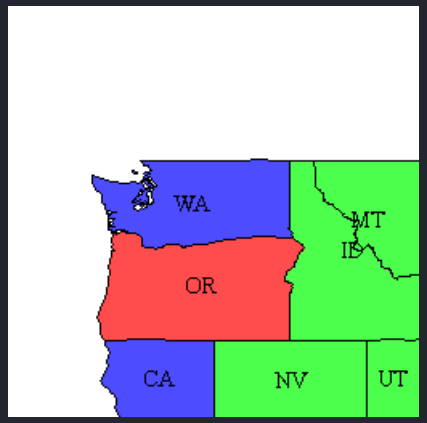
- Antialiasing comes with a high cost, both CPU and size wise (see next slides)
- Might be a good idea to turn it off in resource constrained environments (network/cpu)

opengeo.org

#2: Test outputs (non aa)



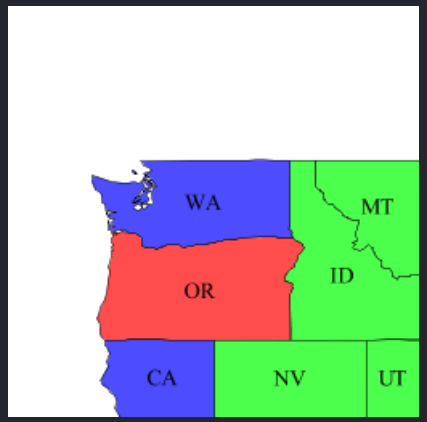
GeoServer (3.3 KB)



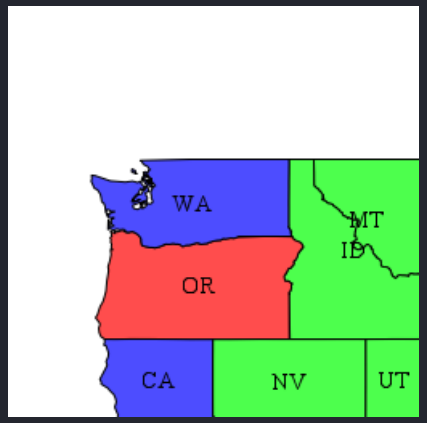
MapServer (2 KB)

- Linework and filling area basically identical
- MapServer could use some more attention labelling (being worked on, the fix will be released in MapServer 5.4)
- Both images are very small, MapServer shows better compression rate

#2: Test output (antialiased)



GeoServer (9.6 KB)



MapServer (13 KB)

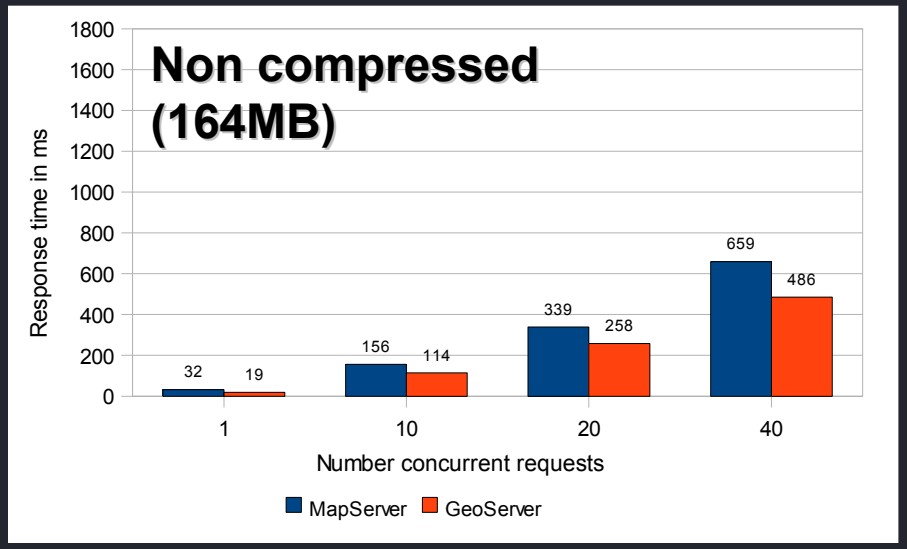
- GeoServer lines too thin for 1 pixel width (being worked on, fix will be released in version 1.7.1)
- GeoServer shows better compression ratio on antialiased data
- No matter what server is being considered, the output size is 4-6 times larger than non antialiased case

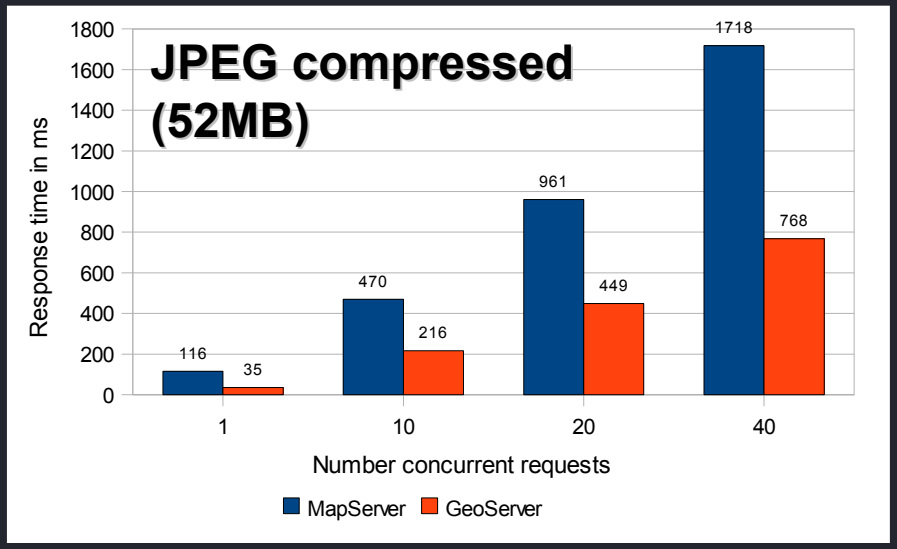
#3: Single GeoTIFF

- A single GeoTIFF file, 6800x6000:
  - Tiled, embedded overviews
- Extracting 256x256 tiles, using 1300+ different bboxes to make sure there is no caching

opengeo.org

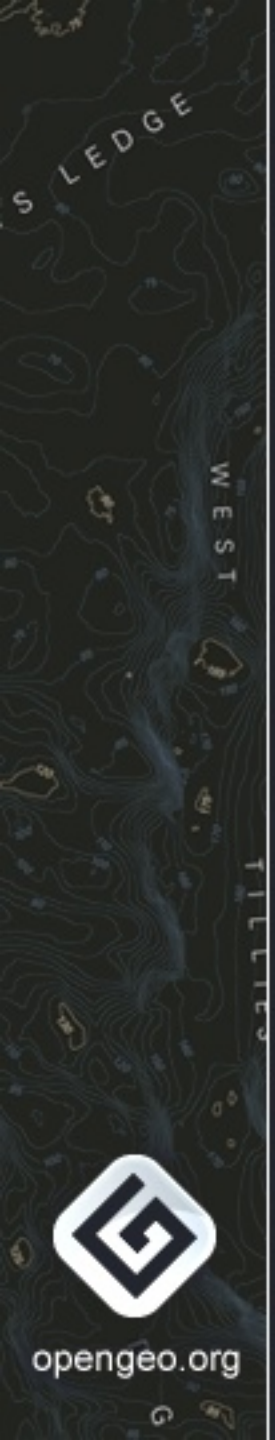
#3: performance results





- JPEG compressing tiles reduces space consumption by 3 times
- But, it slows down decoding visibly (much more so for MapServer)

opengeo.org



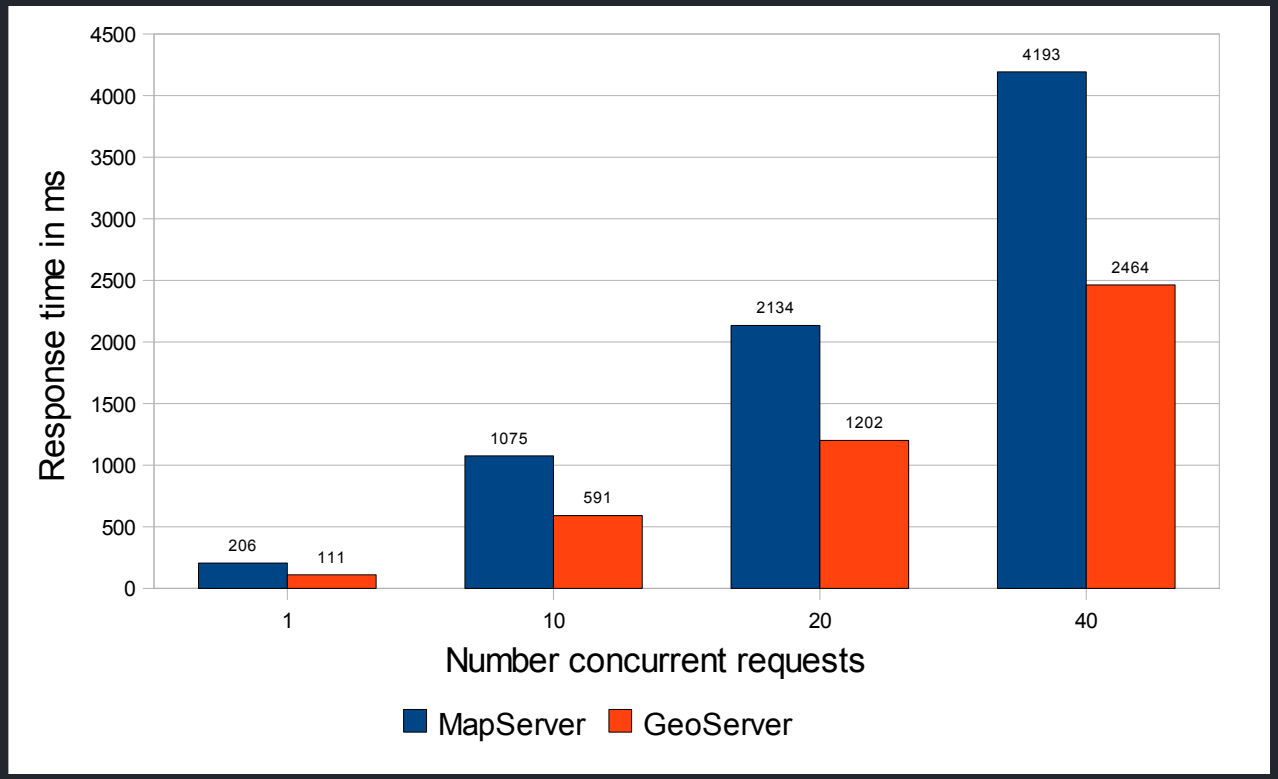
#4: MrSID data

- Landsat, 47637x39167, 206MB, covering the southern part of South Africa
- https://zulu.ssc.nasa.gov/mrsid/
- Both servers using MrSID SDK 3.3 and GDAL
- Extracting 256x256 tiles, jpeg compressed, using 1300+ different bboxes to make sure there is no caching

#4: a look at the data

opengeo.org

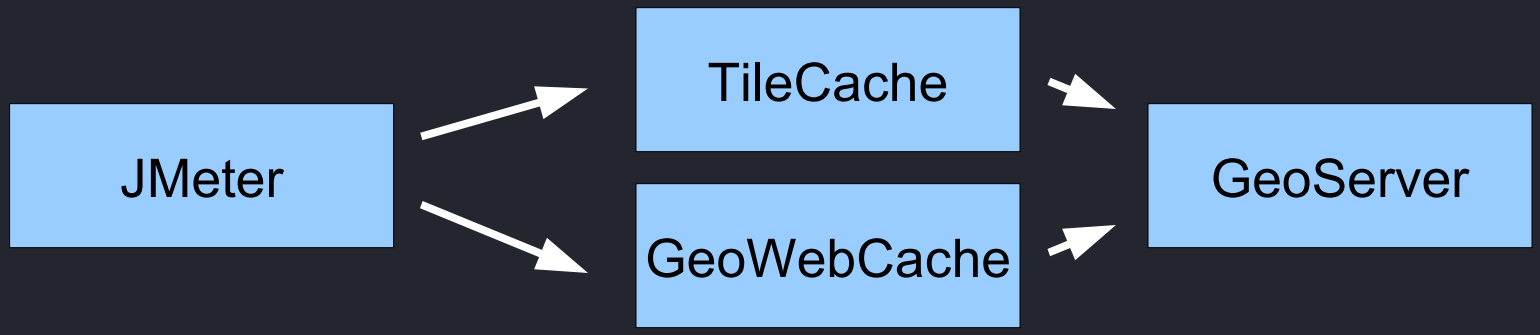
#4: performance results



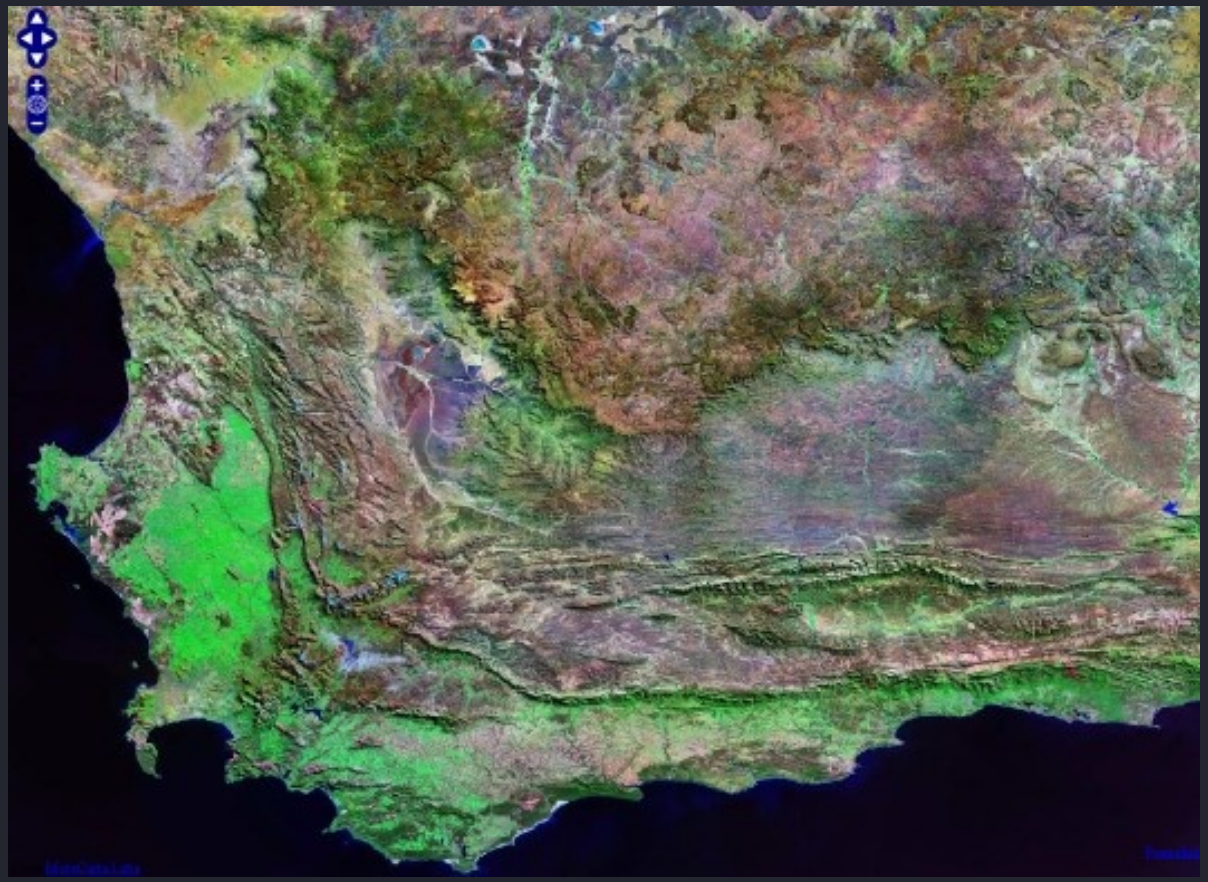
- HP: MapServer is opening the MrSID file for each request, GeoServer is keeping it open (opening and reading MrSID metadata is expensive)

#5: Tile caches

- Testing
  - TileCache 2.04 + mod_python
  - GeoWebCache 1.0-beta0
- Same backend, GeoServer 1.7.0, WMS requests
- Dataset: the MrSid file of the previous test
- Tiles reprojected to WGS84 (jpeg encoding)
- Tile caches are pre-seeded

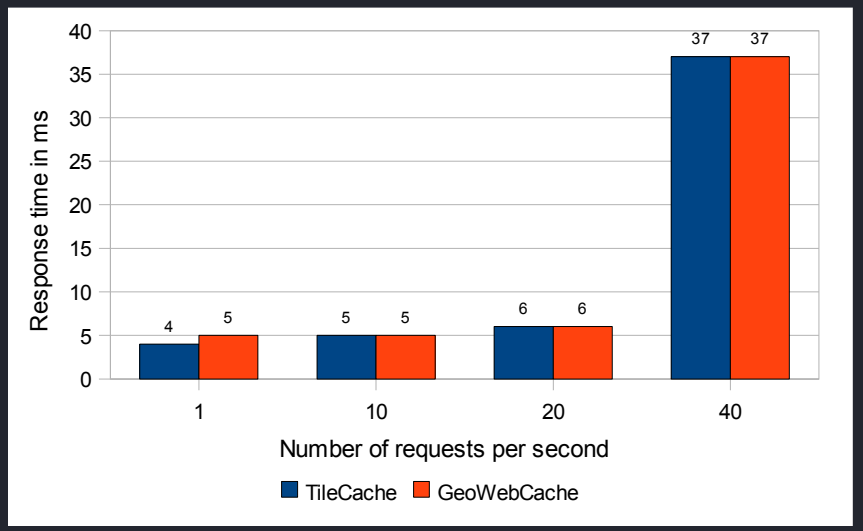


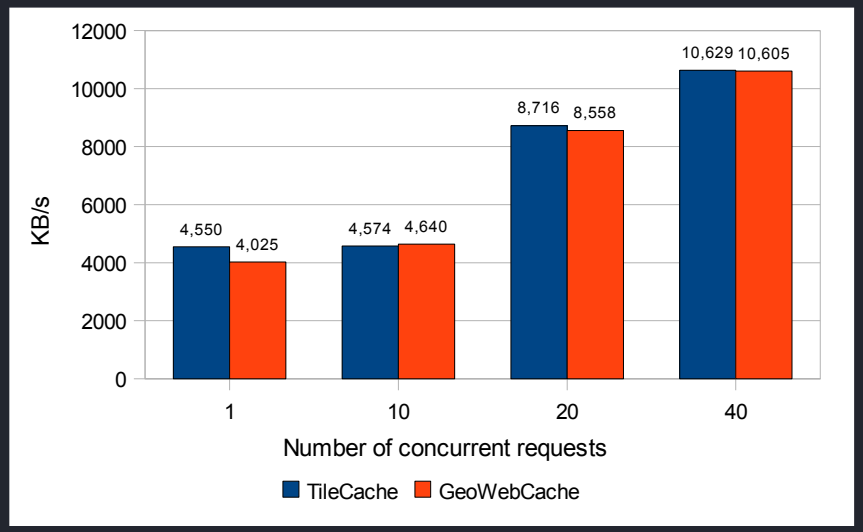
#5: a look at the data



opengeo.org

#5: performance results

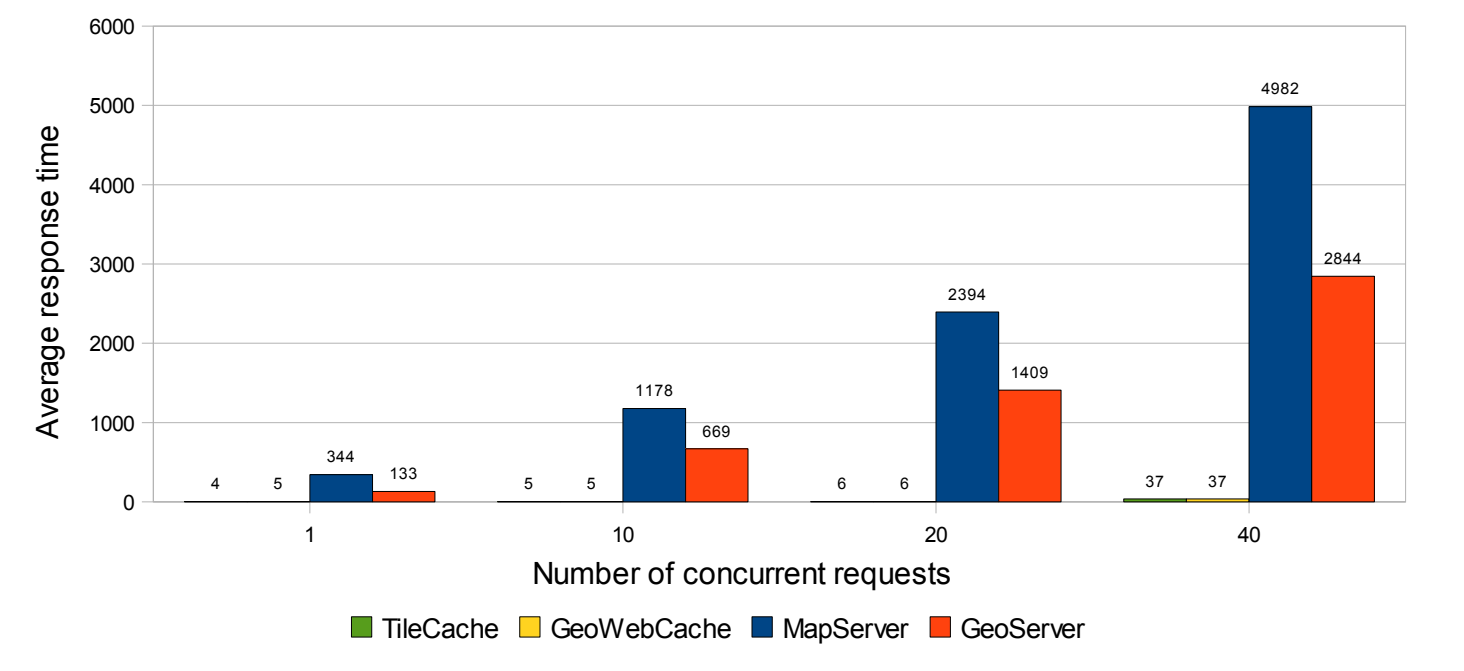




- Virtually identical results, 450+ tiles per second
- The 100MBit connection is saturated, running the test direclty on the server delivers a throughput of over 17MB/s

opengeo.org

#5: quantifying the tile caching gain

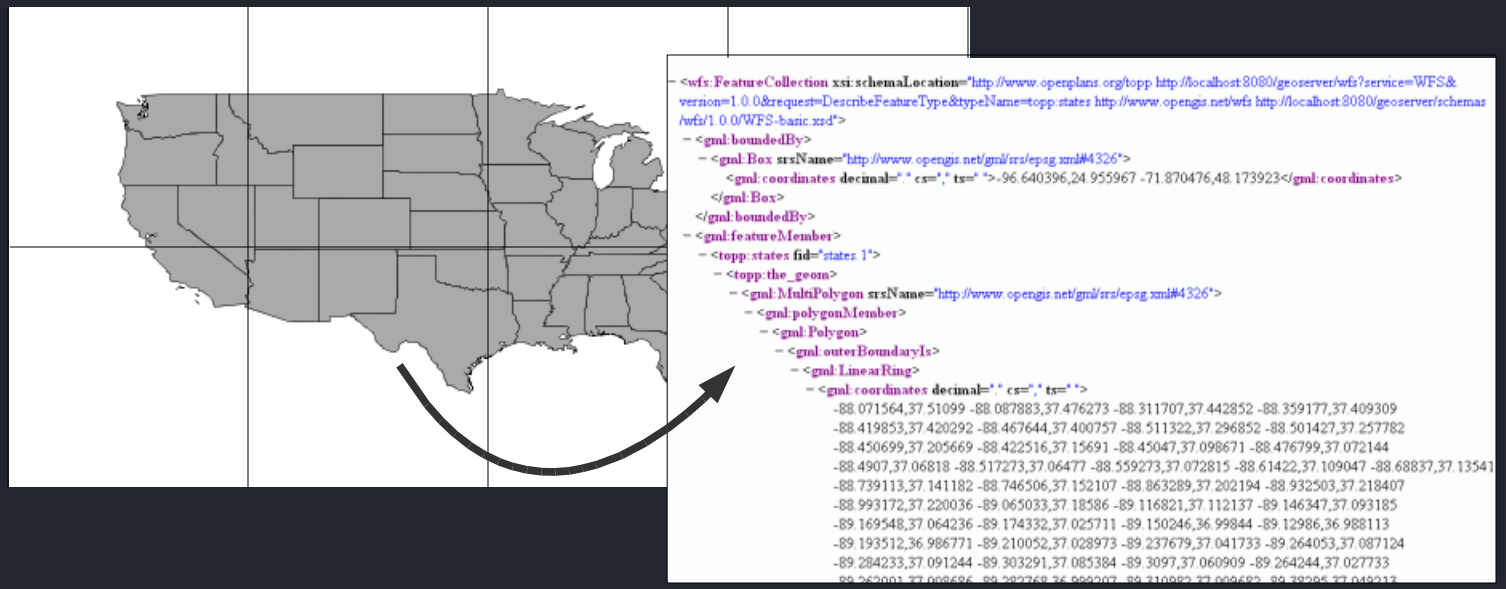


- Made the same requests against MapServer and GeoServer un-cached
- In this case, the benefit is around 100 times
- The gain won't be as much if you consider simple vector rendering

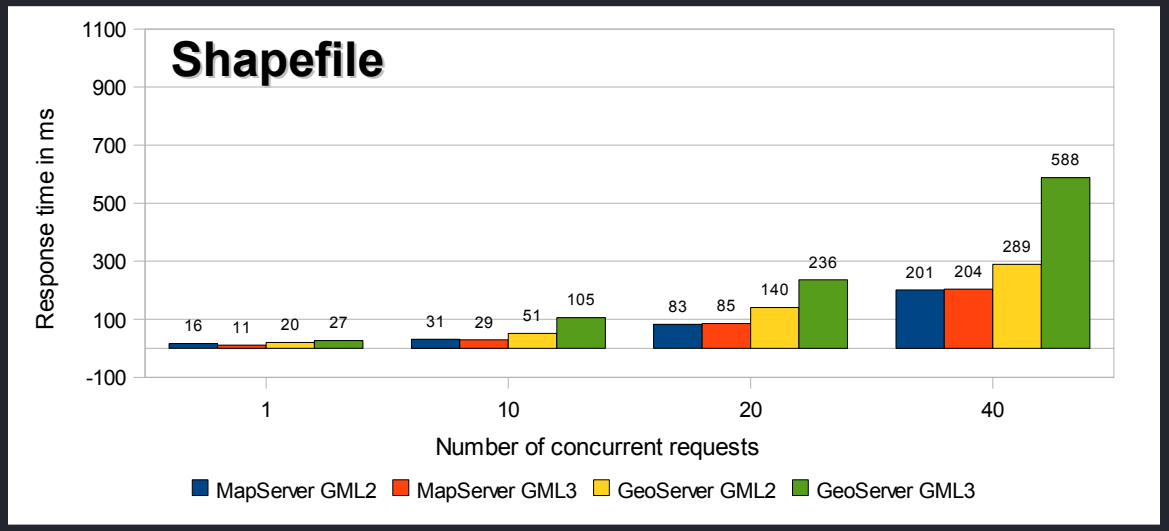
opengeo.org

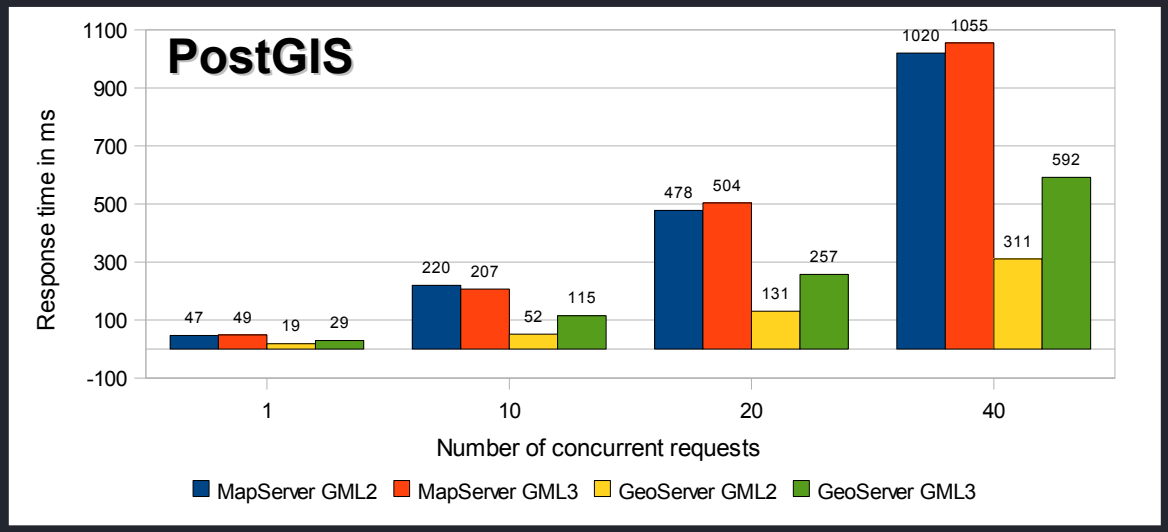
#6: WFS requests

- WFS requests with BBOX filtering
- Using the same tiles and data as the thematic map benchmark
- Shapefile and PostGIS backends
- WFS 1.0 and WFS 1.1 GetFeature



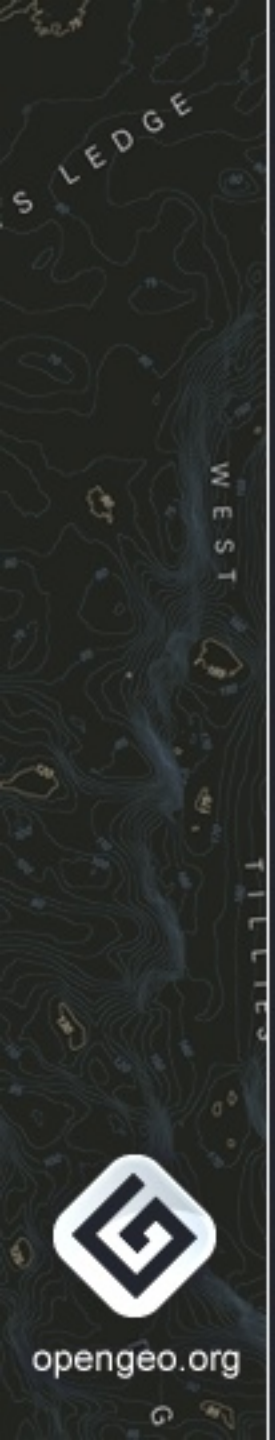
#6: performance results





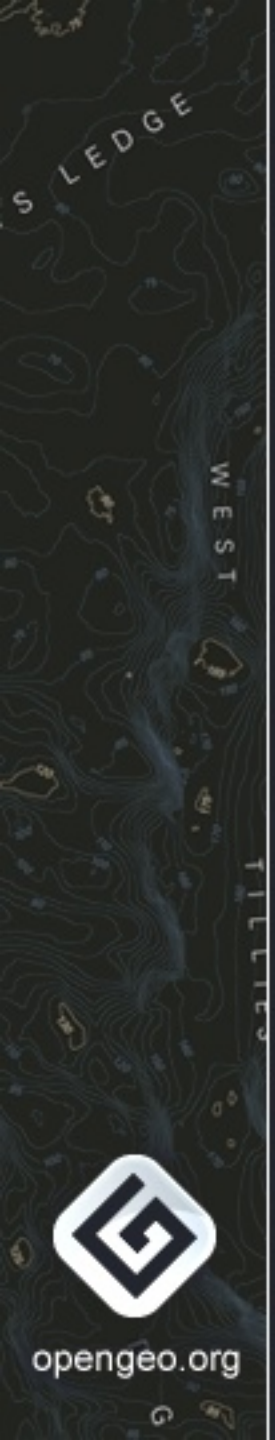
- MapServer is very fast with shapefiles, not as much with PostGIS
- GeoServer GML3 output could use some work

opengeo.org



Summary

- Both servers are very fast (most of the test deliver more than 50 responses per second... on a notebook!)
- But proper configuration is crucial to get good results
- As we have seen, both server can still work to get better, as the relative differences in results do show



QUESTIONS?