WMS Performance Shootout

Andrea Aime (GeoServer)
Jeff McKenna (MapServer)

Executive summary

- Compare the performance of WMS servers
  - GeoServer
  - MapServer
- In a number of different workloads:
  - Vector: plain polygon, point with graphical icons, road network with cased lines, road network with labels
  - Raster: global imagery
- Against different data backends:
  - Vector: shapefiles, PostGIS, Oracle spatial, SDE over Oracle spatial
  - Raster: ECW and GeoTiff mosaics

Benchmarking History

- 3rd FOSS4G benchmarking exercise. Past exercises included:
  - FOSS4G 2007: Refractions Research run and published the first comparison with the help of GeoServer and MapServer developers. Focus on big shapefiles, postgis, minimal styling
  - FOSS4G 2008: OpenGeo run and published the second comparison with some review from the MapServer developers. Focus on simple thematic mapping, raster data access, WFS and tile caching
- Friendly competition: goal is to improve all software

Past MapServer improvements

- improvements in large shapefile indexing
- raster read optimization (single pass for multiple bands)
- enhancing polygon labelling placement
- EPSG codes caching
- PostGIS improvements
- Oracle improvements

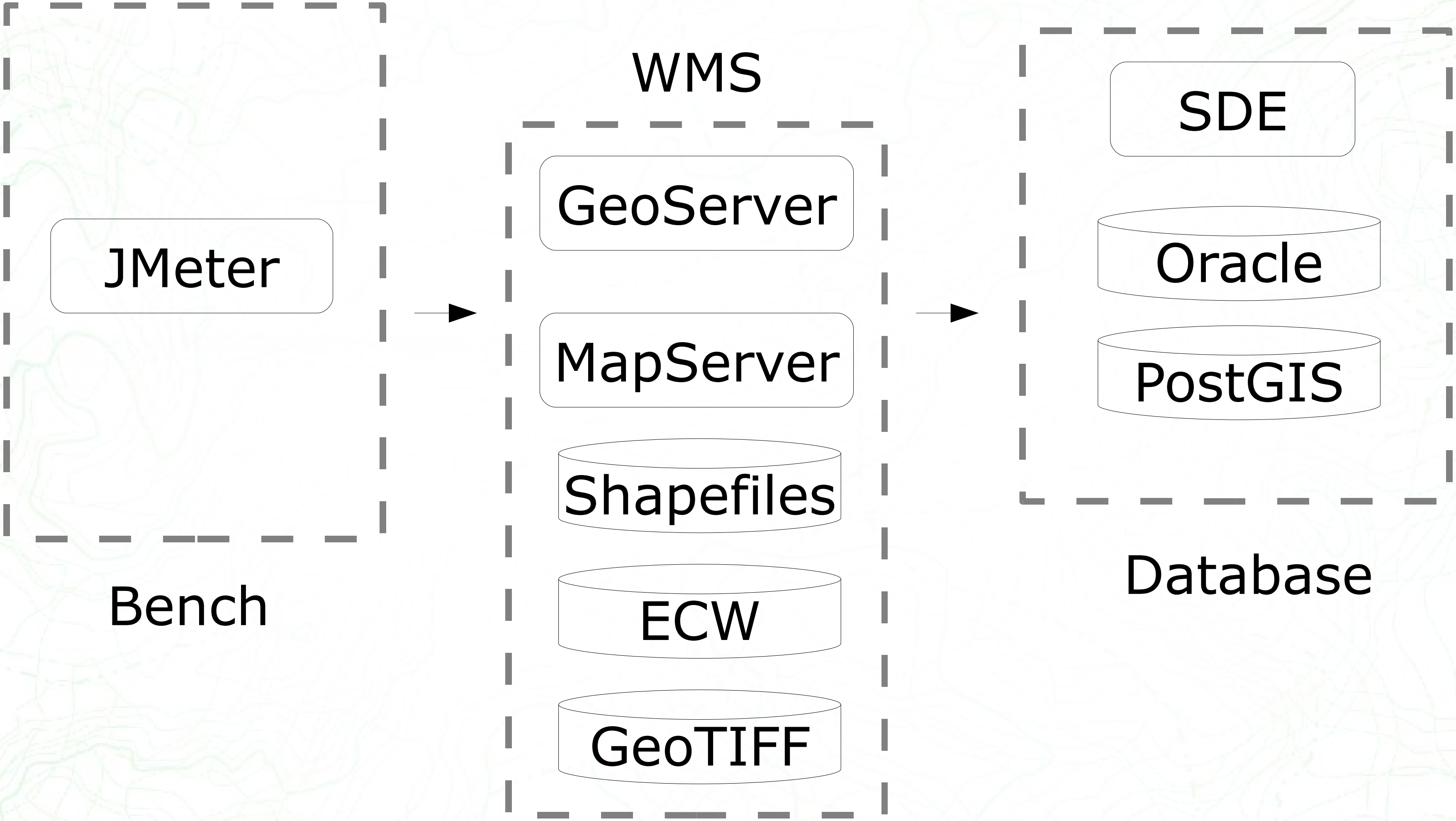
Past GeoServer improvements

- Overall rendering pipeline improvements
- GML writing speed-ups
- Raster data access improvements
- PostGIS data store optimizations

Rules of engagement

- Each server is tested in its latest version
- Each server performs exactly the same workload
  - Same set of WMS requests
  - Same data backends
  - Same image output format
- All modifications made to improve performance are to be included in a future release of the software
- Data used cannot be modified for display, other than indexing
- All testing to be done on the same benchmarking machines

Hardware Configuration



Hardware specs

- Bench (2003):
  - Dell PowerEdge 1750
  - 1 Xeon Nocona 3Ghz, 1 GB RAM, OS: RedHat Enterprise 5.3
- WMS (2004):
  - Dell PowerwEdge 2850
  - 4 Xeon Nocona 3.4Ghz, 8 GB RAM
  - 6 73 GB, 15K RPM hard drives, OS: RedHat Enterprise 5.3
- Database (2007):
  - Dell Optiplex 755 tower
  - 1 x Intel Core2 Duo CPU E6750 @ 2.66GHz / Dual Core, 4Gb RAM
  - 100Gb SATA 3.0Gb/s 7200 rpm HD, OS: RedHat Enterprise 5.3

Hot vs Cold Benchmarks

- Hot benchmark
  - The file system caches are fully loaded
  - The software has had the occasion to perform whatever pre-processing is necessary (e.g., open connections to the data source)
- Cold benchmark
  - The system has been just started
  - There is no existing cache
- Both are unrealistic, production is usually a mix of both
- Hot benchmarks are more practical to run

Methodology

- Each test run performs requests with 1, 10, 20 and 40 parallel clients (for a total of 1500 requests)
- Each test uses a random set of requests stored in a CSV file: no two requests in the same run are equal, but all servers perform the same workload
- For each request the random factors are:
  - The image size (between 640x480 and 1024x768)
  - The geographic envelope (extent)
- Each test is run three times in a row, the results of the third run are used for the comparison: this benchmark assumes full file system caches (“hot” benchmark)
- The other GIS server is shut down while the tests are run

Datasets Used

- **Polygon layer: areawater_merge**: the TIGER 2008 set of polygons describing water surfaces for the state of Texas. 380000 polygons, 195MB shapefile, EPSG:4269
- **Point layer: gnis_names09**: all location and points of interest names for the state of Texas in the GNIS database. 103000 points, 2.8 MB shapefile, EPSG:4269
- **Line layer, edges_merge**: all line elements (rivers and roads) from the TIGER 2008 dataset for the state of Texas. over 5M lines, 1.2GB shapefile, 1.4GB dbf, EPSG:4269
- **Raster layer**: Bluemarble TNG, 86400 x 43200 pixels, 7 overview layers (TIF, ECW, tiled 512&8, GeoRaster)

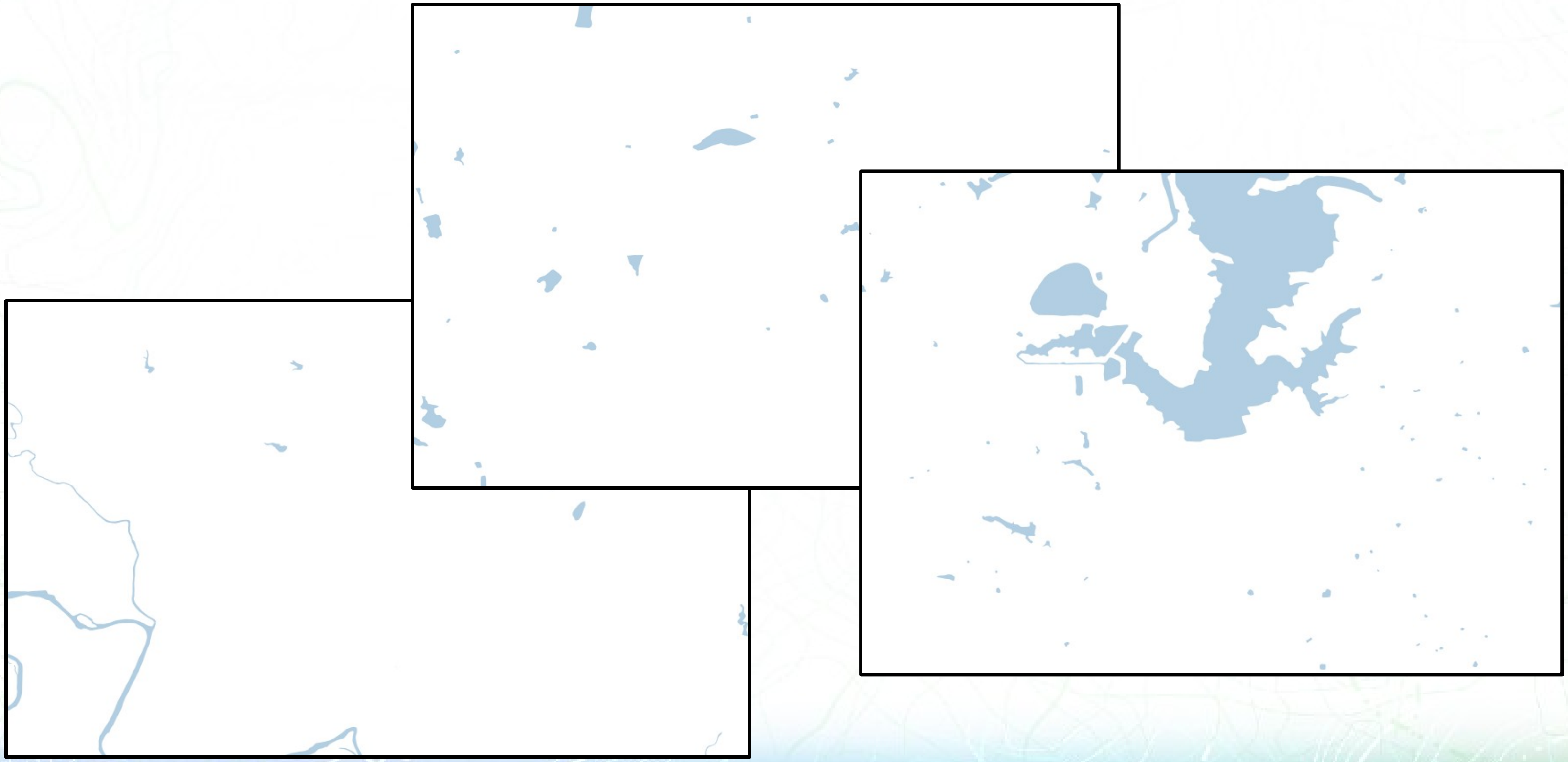
Equivalent traffic table

- The presentation charts results as throughput in requests per second.
- This table translates hits per second **at sustained peak load** in terms of requests per hour and per day

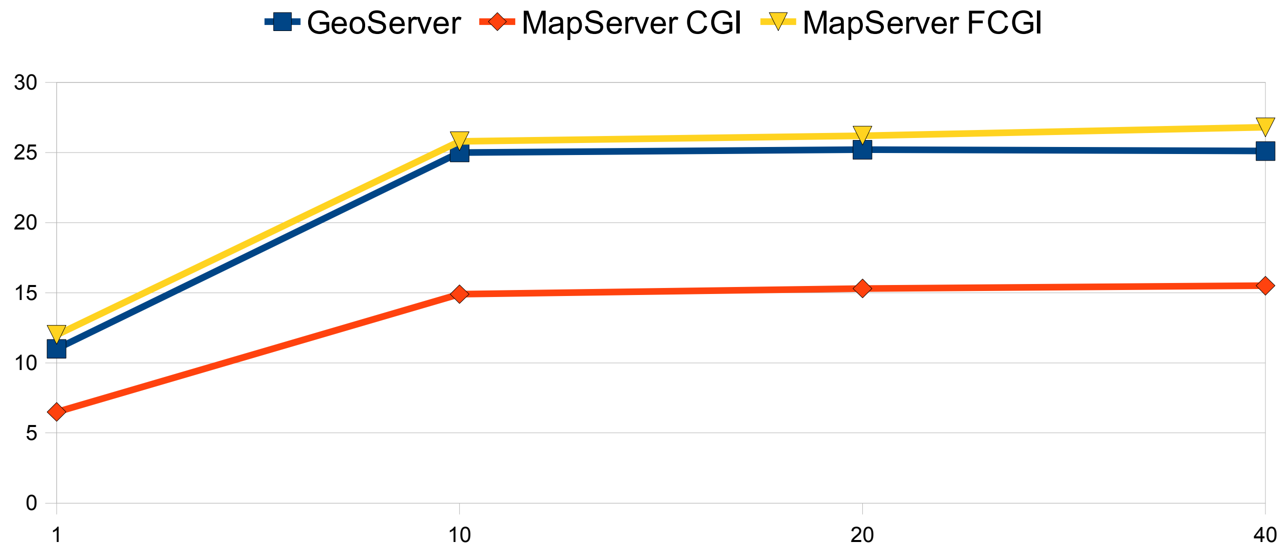
Hits/s	Hits/hour	Hits/day
5	18,000	432,000
10	36,000	864,000
15	54,000	1,296,000
20	72,000	1,728,000
25	90,000	2,160,000

T1: plain polygon rendering

- TIGER 2008 “areawater” layer, merged over Texas
- Simply fill the polygon in blue, no outline
- Image scales roughly between 1:10.000 and 1:100.000

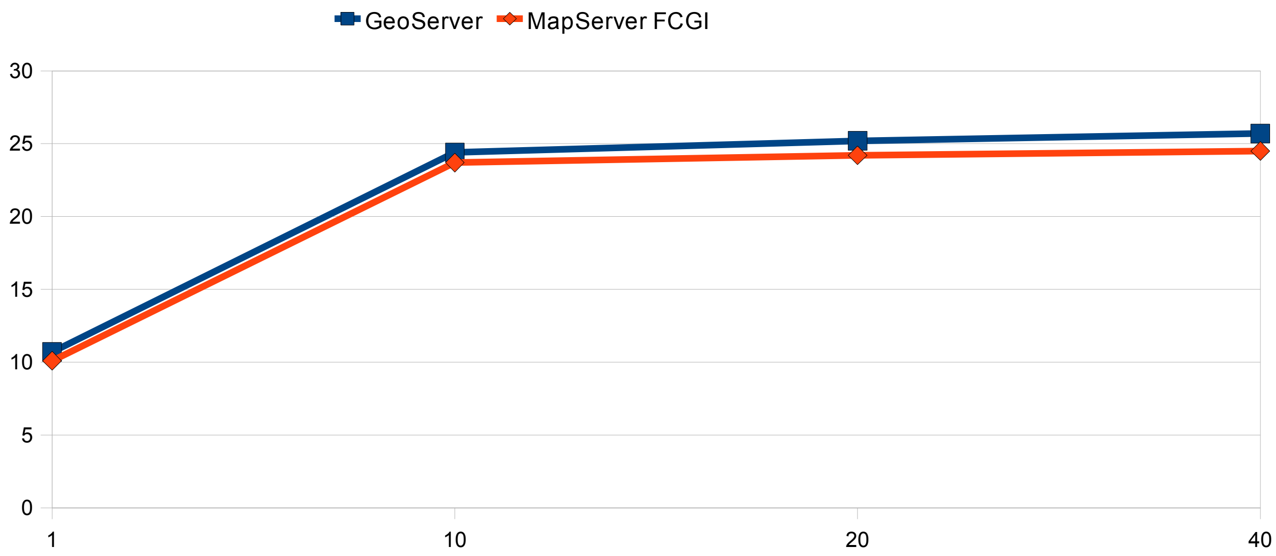


T1: Shapefiles



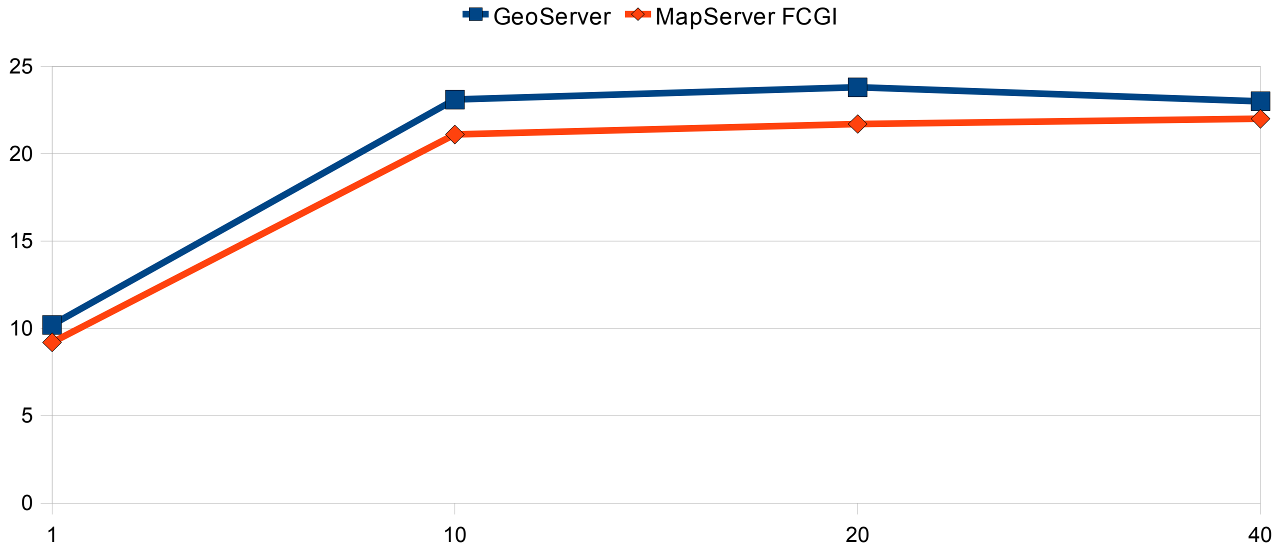
Threads	GeoServer	MapServer CGI	MapServer FCGI
1	11	6.5	12
10	25	14.9	25.8
20	25.2	15.3	26.2
40	25.1	15.5	26.8

T1: PostGIS



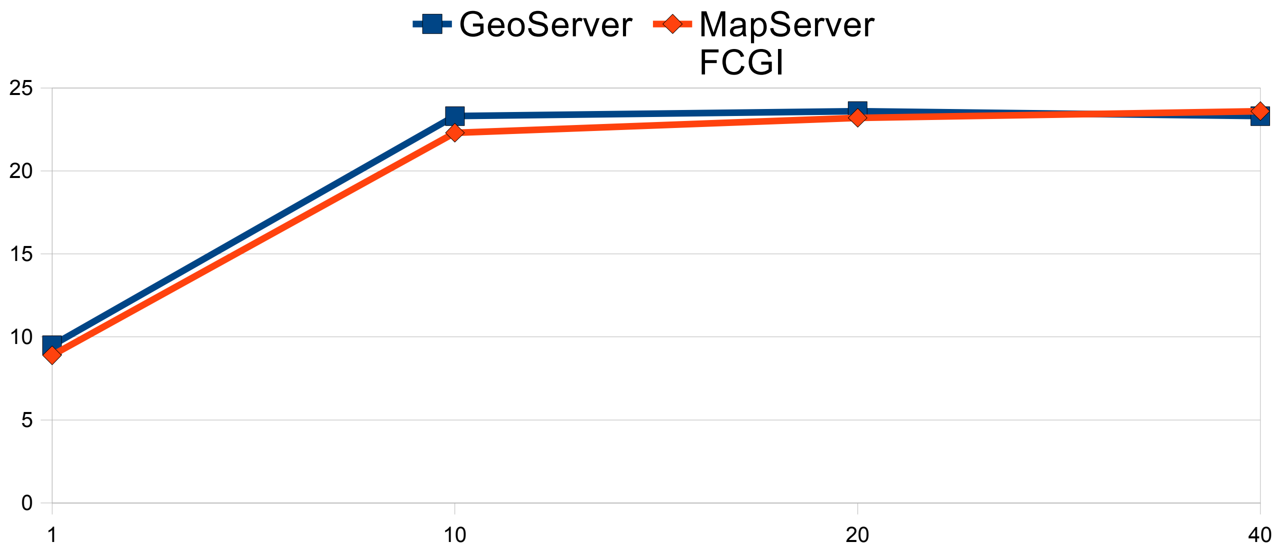
Threads	GeoServer	MapServer FCGI
1	10.7	10.1
10	24.4	23.7
20	25.2	24.2
40	25.7	24.5

T1: Oracle



Threads	GeoServer	MapServer FCGI
1	10.2	9.2
10	23.1	21.1
20	23.8	21.7
40	23	22

T1: SDE



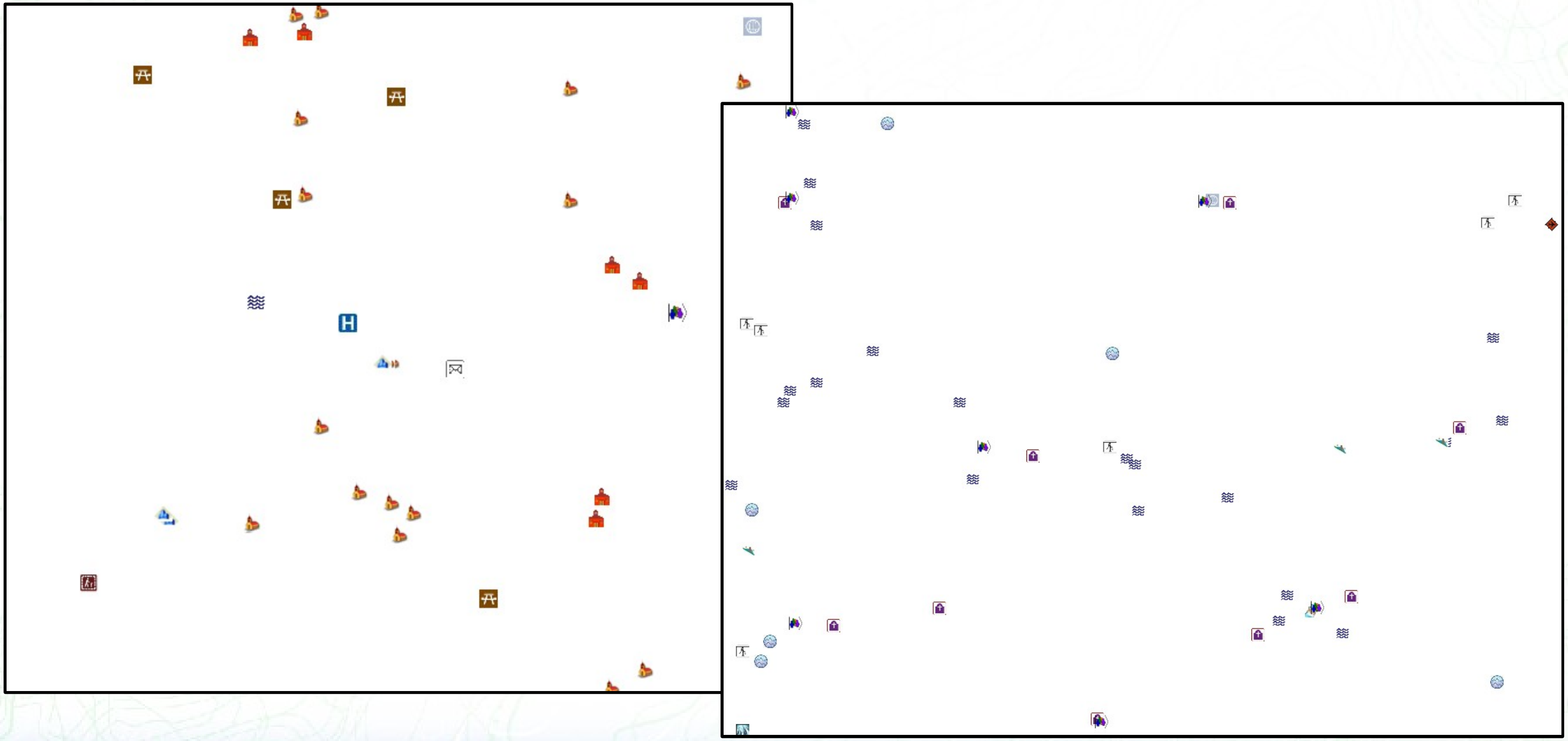
Threads	GeoServer	MapServer FCGI
1	9.5	8.9
10	23.3	22.3
20	23.6	23.2
40	23.3	23.6

T1: observations

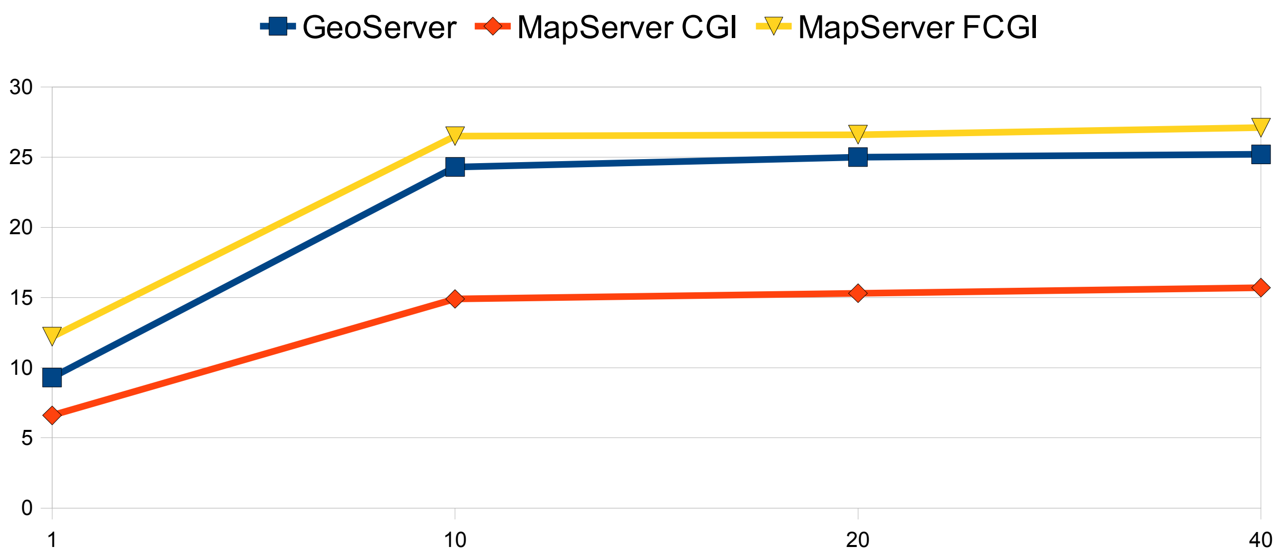
- GeoServer is between 25% and 50% faster reading shapefiles out of the box, MapServer with FastCGI is slightly faster
- PostGIS performs roughly 5% better than Oracle on higher loads
- SDE seems to introduce no overhead compared to direct Oracle access for GeoServer, and a slight overhead for MapServer
- GeoServer and MapServer have exactly the same performance when the backend is PostGIS and Oracle

T2: point rendering

- GNIS names dataset over Texas
- Each point is rendered using a little GIF/PNG graphics

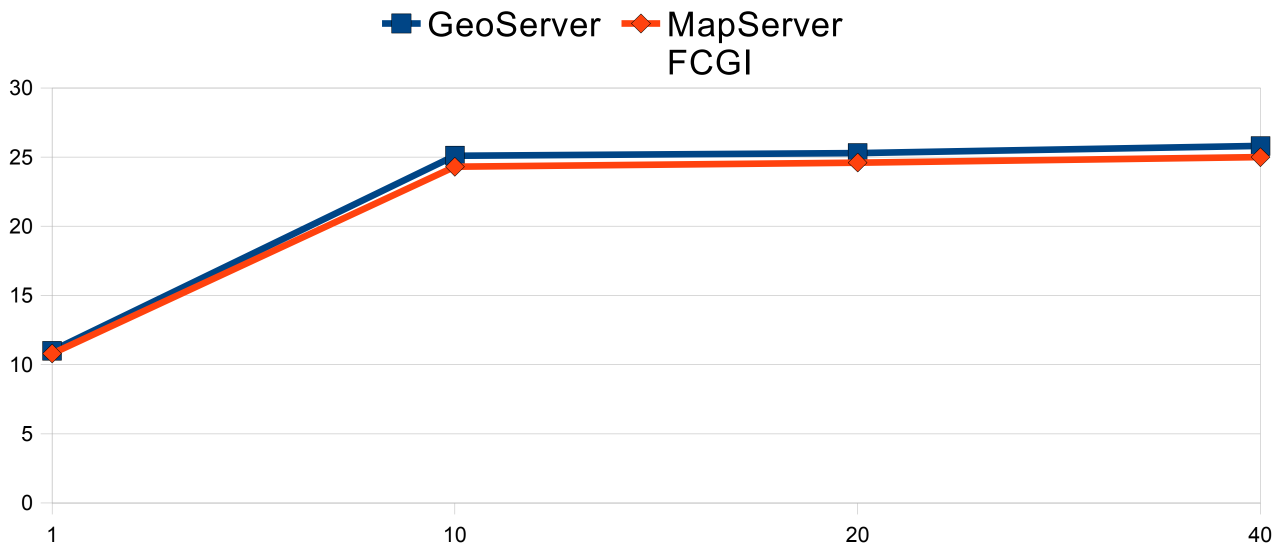


T2: shapefiles



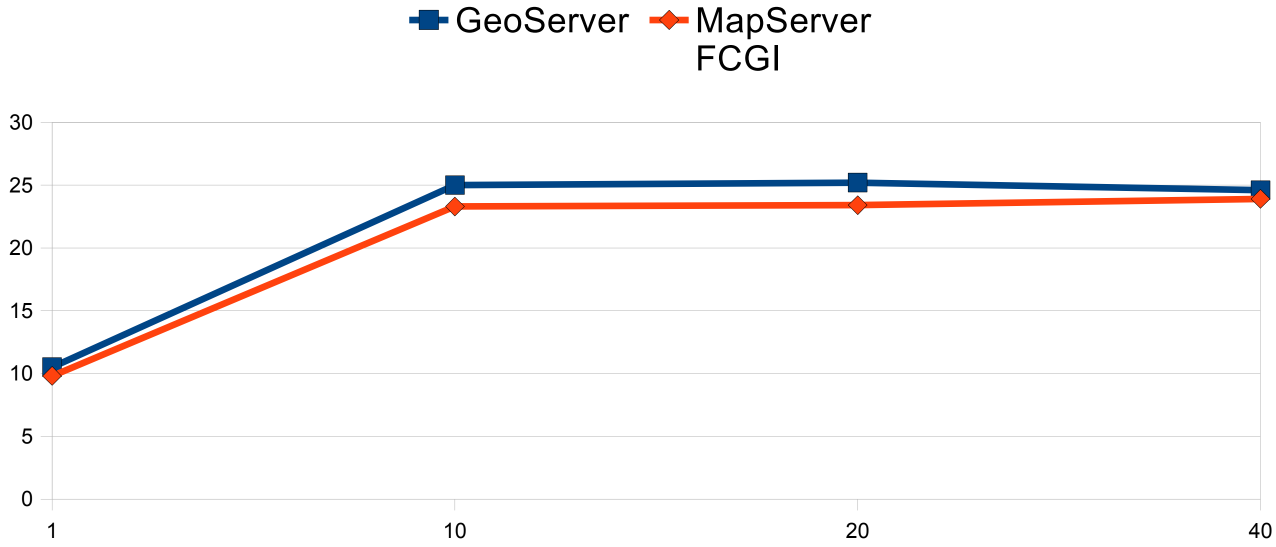
Threads	GeoServer	MapServer CGI	MapServer FCGI
1	9.3	6.6	12.2
10	24.3	14.9	26.5
20	25	15.3	26.6
40	25.2	15.7	27.1

T2: PostGIS



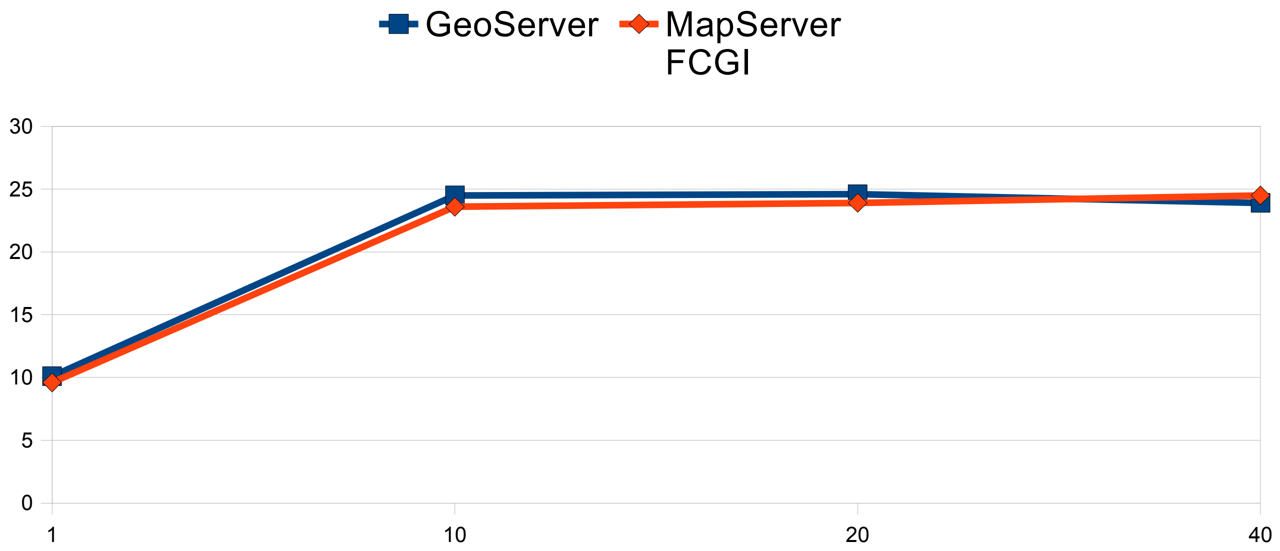
Threads	GeoServer	MapServer FCGI
1	11	10.8
10	25.1	24.3
20	25.3	24.6
40	25.8	25

T2: Oracle



Threads	GeoServer	MapServer FCGI
1	10.5	9.8
10	25	23.3
20	25.2	23.4
40	24.6	23.9

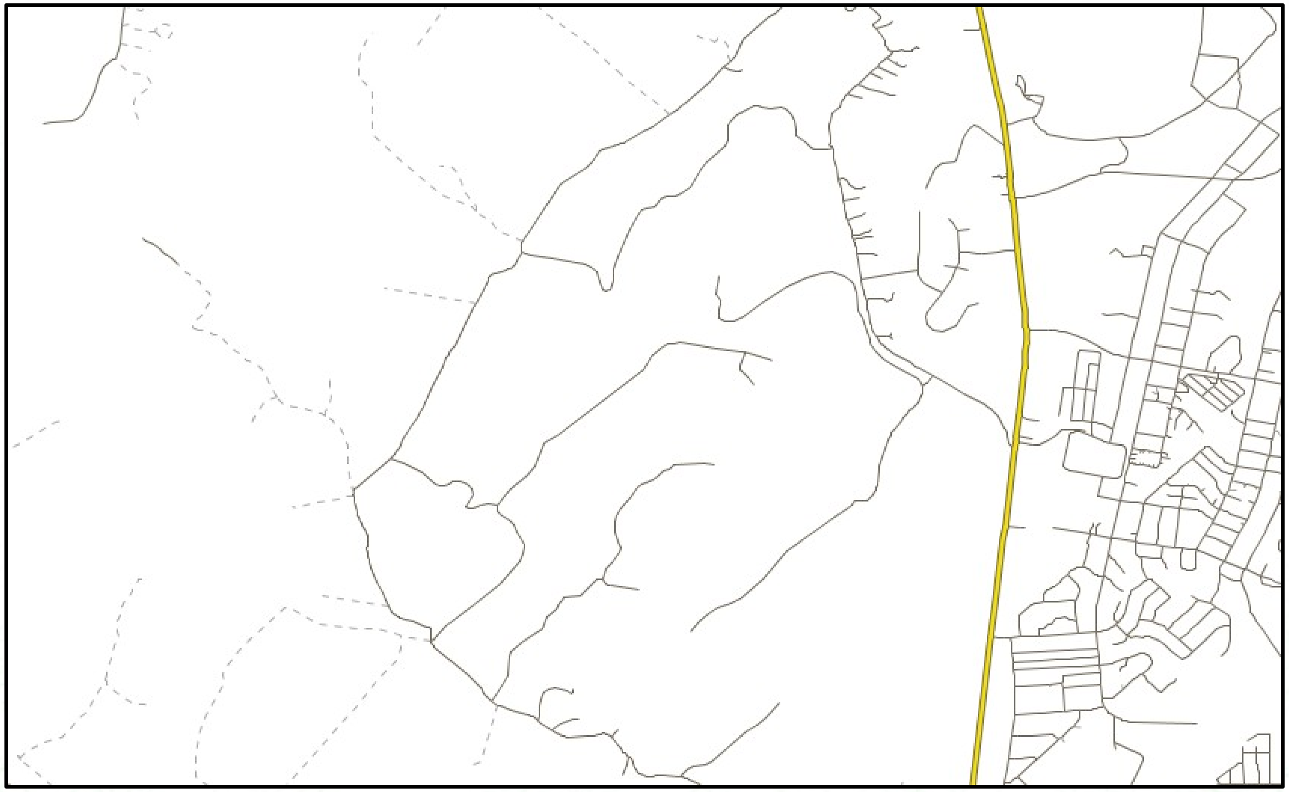
T2: SDE



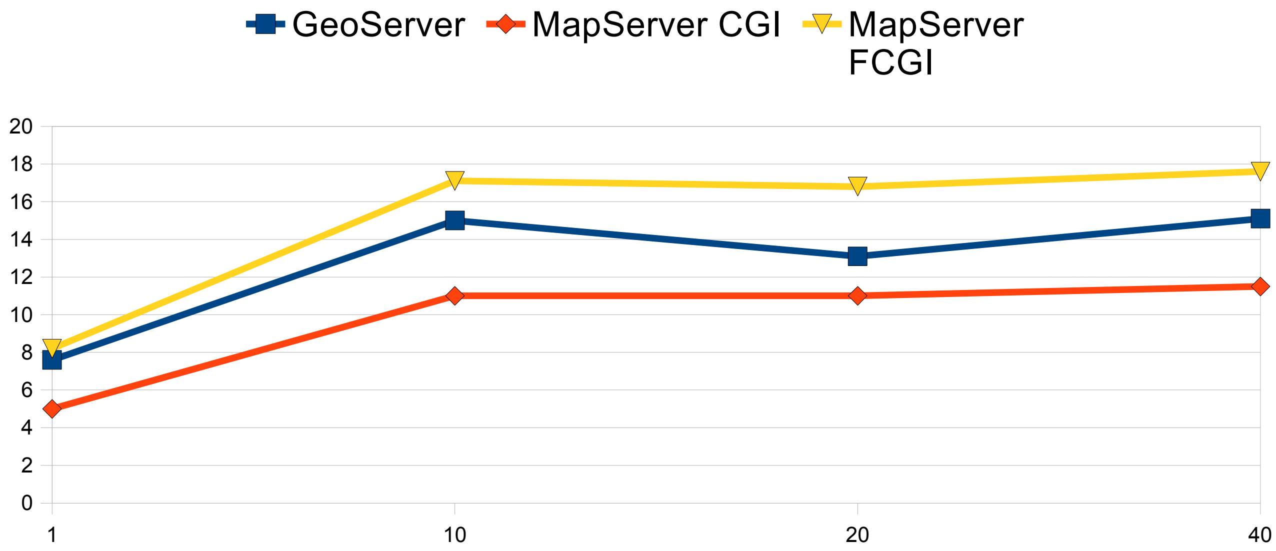
Threads	GeoServer	MapServer FCGI
1	10.1	9.6
10	24.5	23.6
20	24.6	23.9
40	23.9	24.5

T3: roads map

- TIGER 2008 edges, just roads (rivers filtered out), three different line styles based on the road class

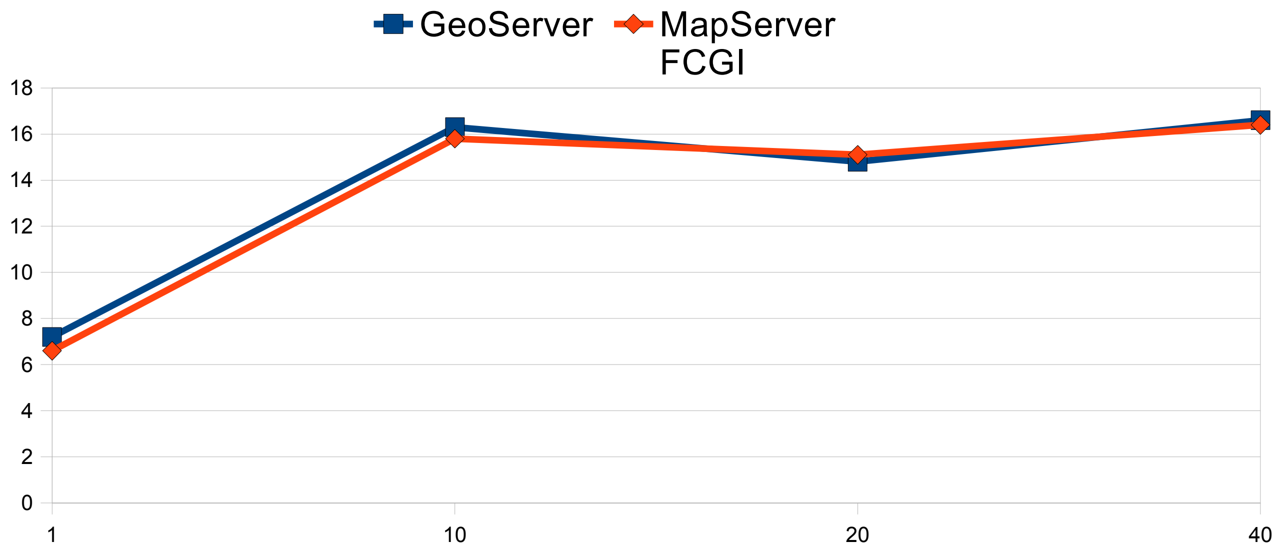


T3: shapefiles



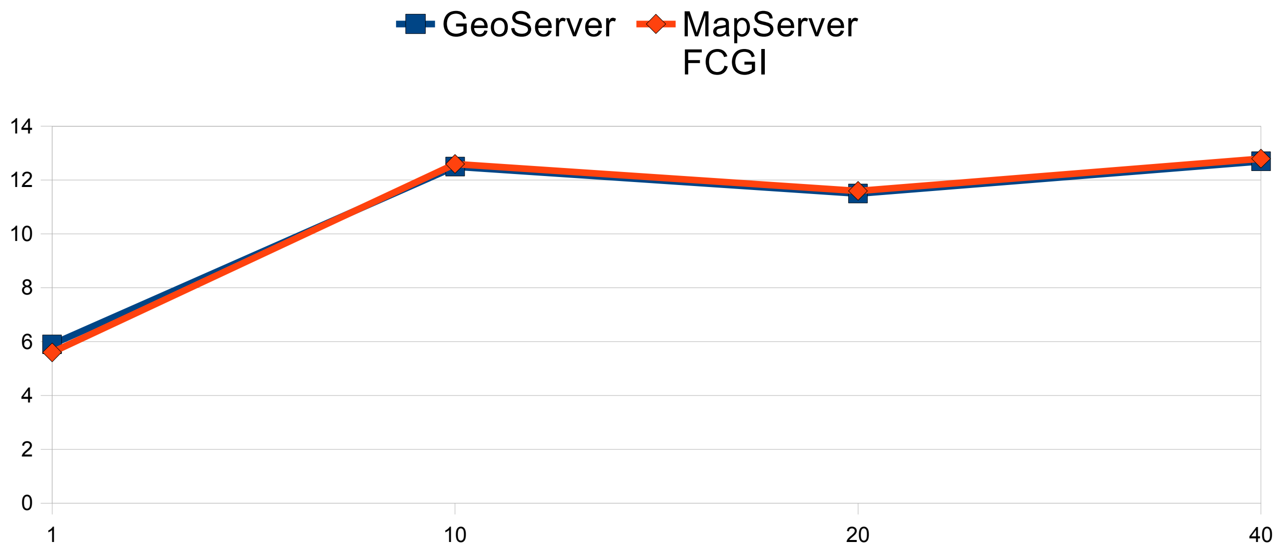
Label	GeoServer	MapServer CGI	MapServer FCGI
1	7.6	5	8.2
10	15	11	17.1
20	13.1	11	16.8
40	15.1	11.5	17.6

T3: PostGIS



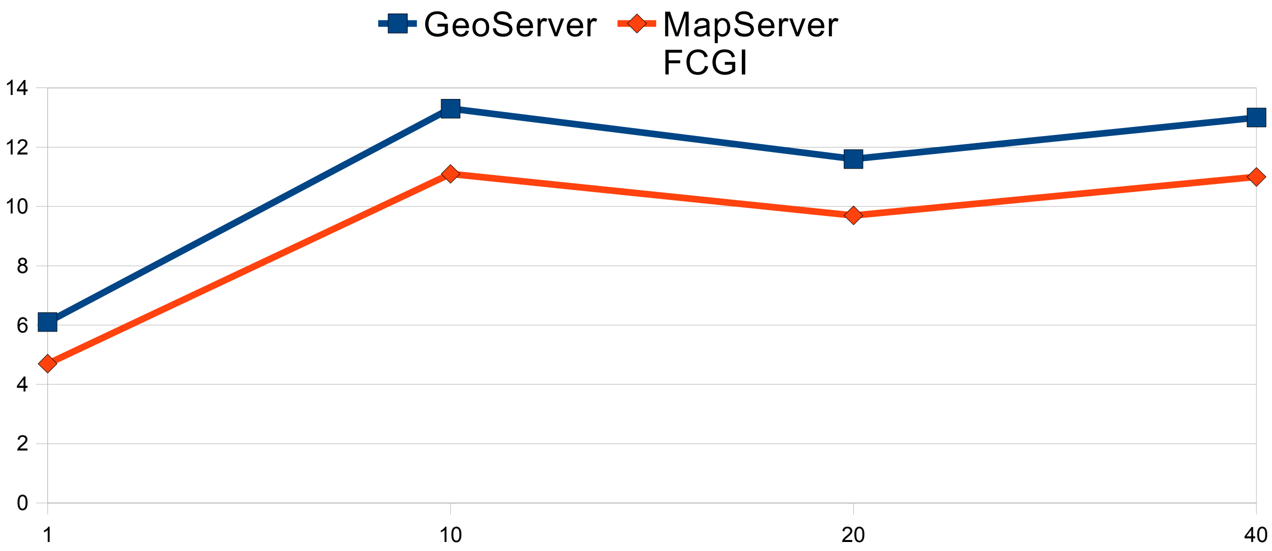
Label	GeoServer	MapServer FCGI
1	7.2	6.6
10	16.3	15.8
20	14.8	15.1
40	16.6	16.4

T3: Oracle



Label	GeoServer	MapServer FCGI
1	5.9	5.6
10	12.5	12.6
20	11.5	11.6
40	12.7	12.8

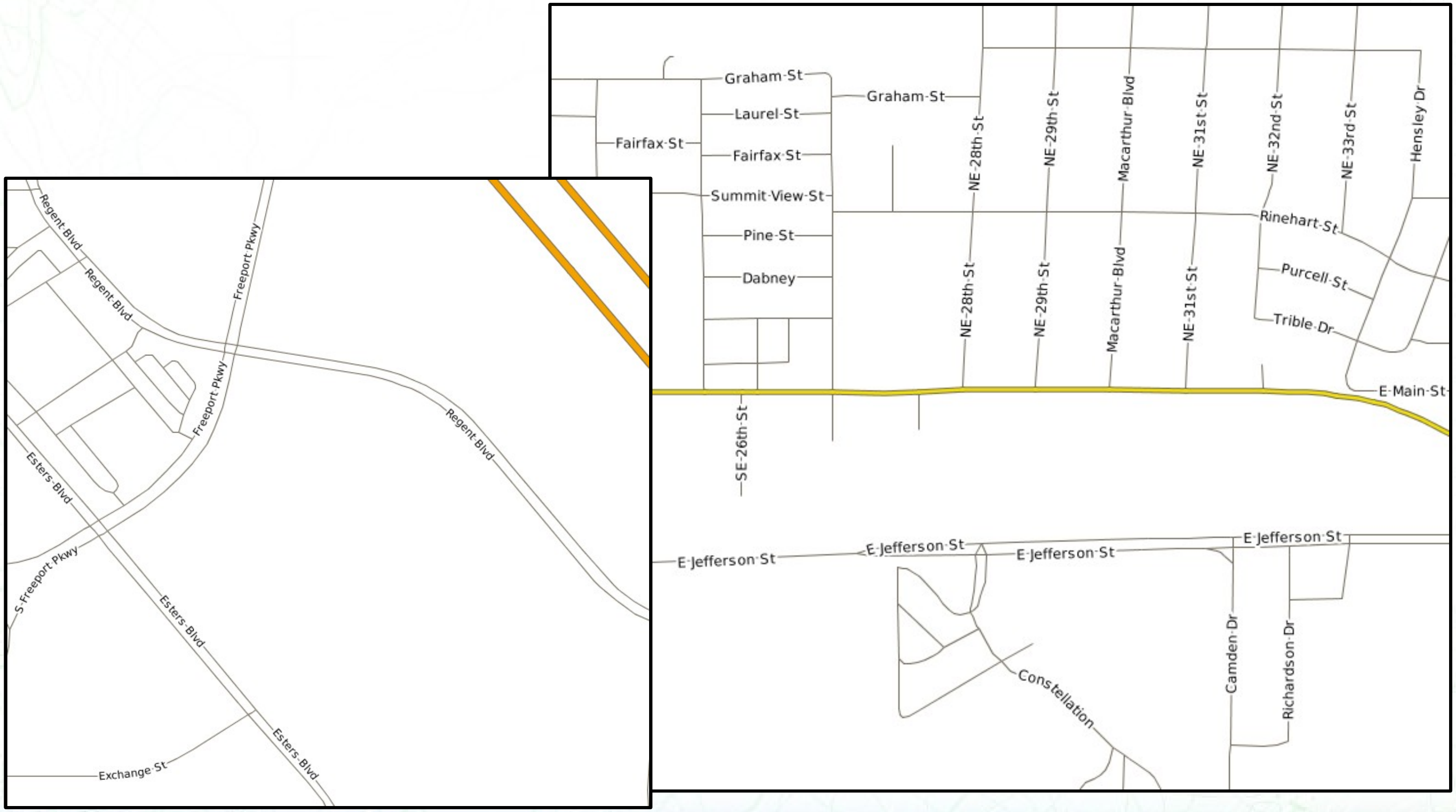
T3: SDE



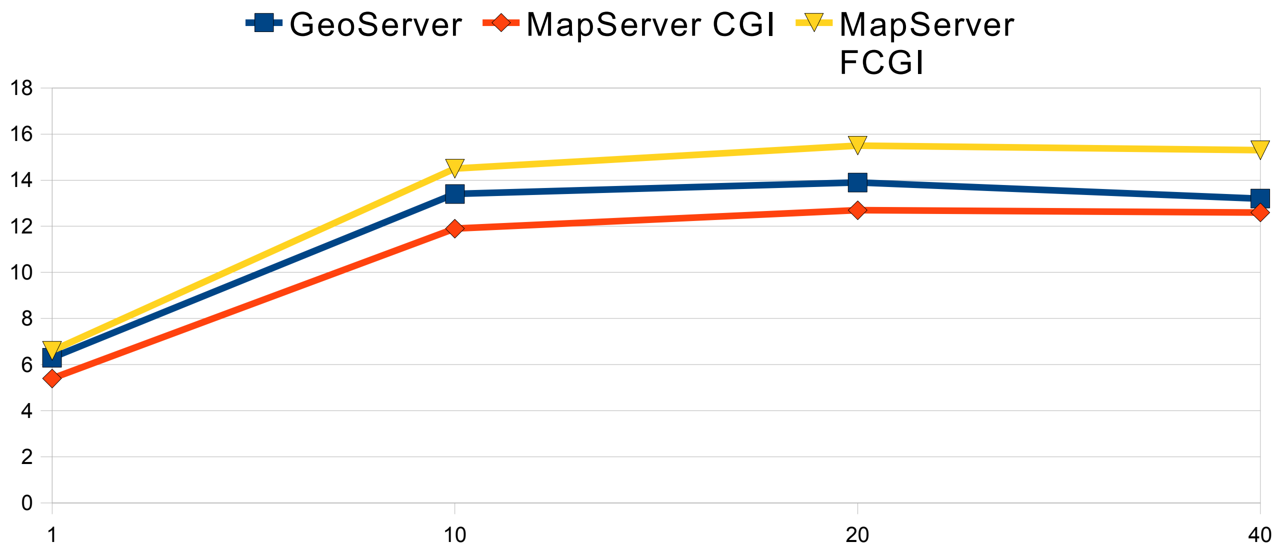
Label	GeoServer	MapServer FCGI
1	6.1	4.7
10	13.3	11.1
20	11.6	9.7
40	13	11

T4: labelled roads

- Same as T3, but higher scale (between 1:10k and 1:3k) and with labels following lines

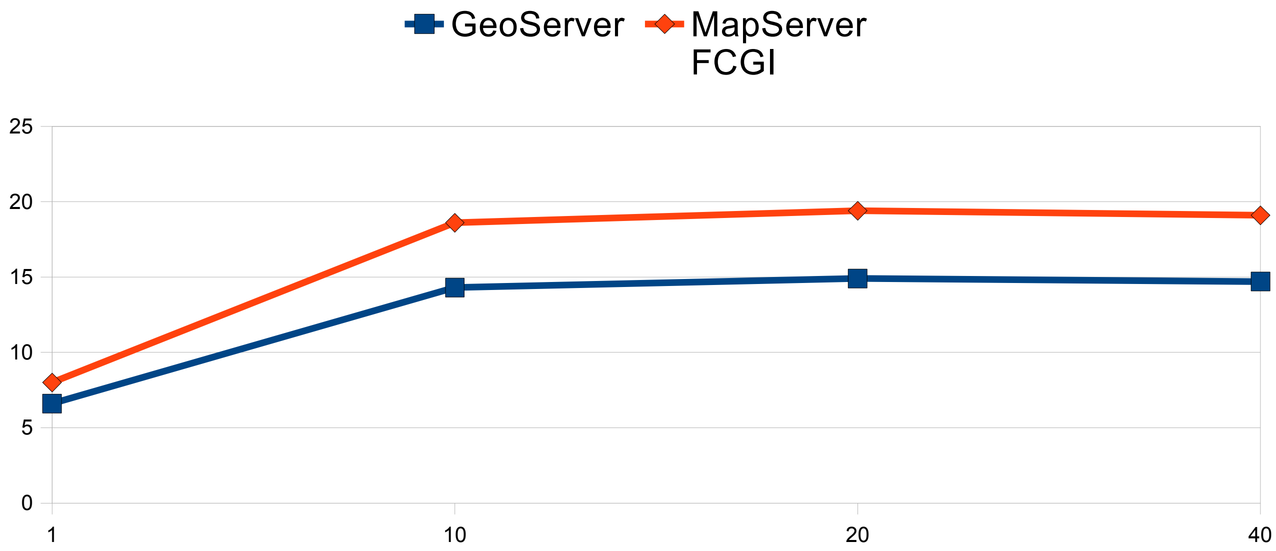


T4: shapefiles



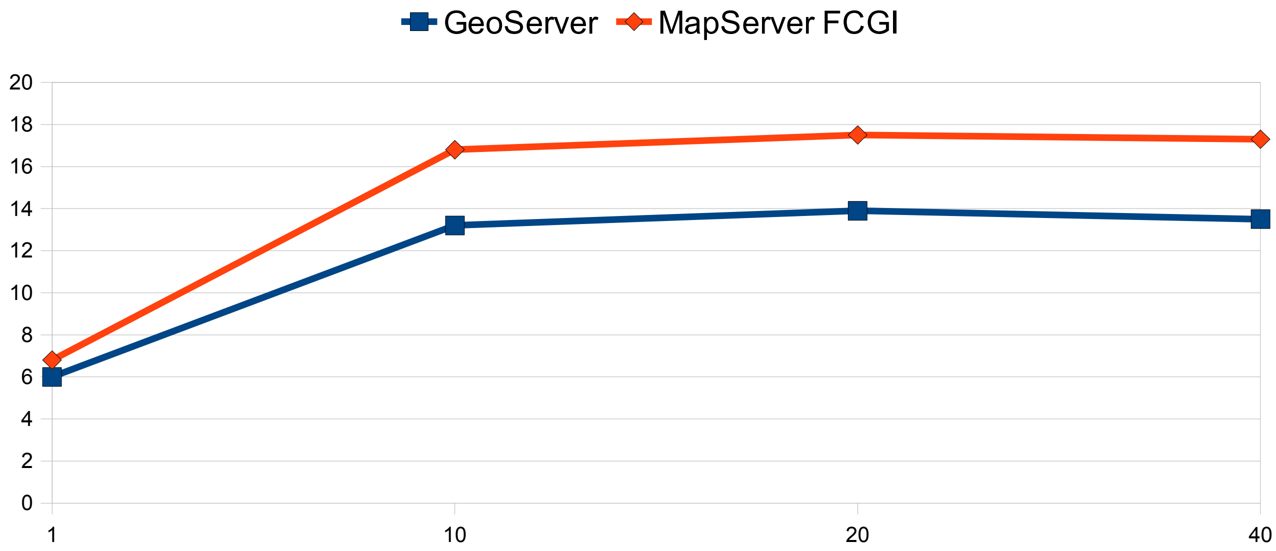
Label	GeoServer	MapServer CGI	MapServer FCGI
1	6.3	5.4	6.6
10	13.4	11.9	14.5
20	13.9	12.7	15.5
40	13.2	12.6	15.3

T4: PostGIS



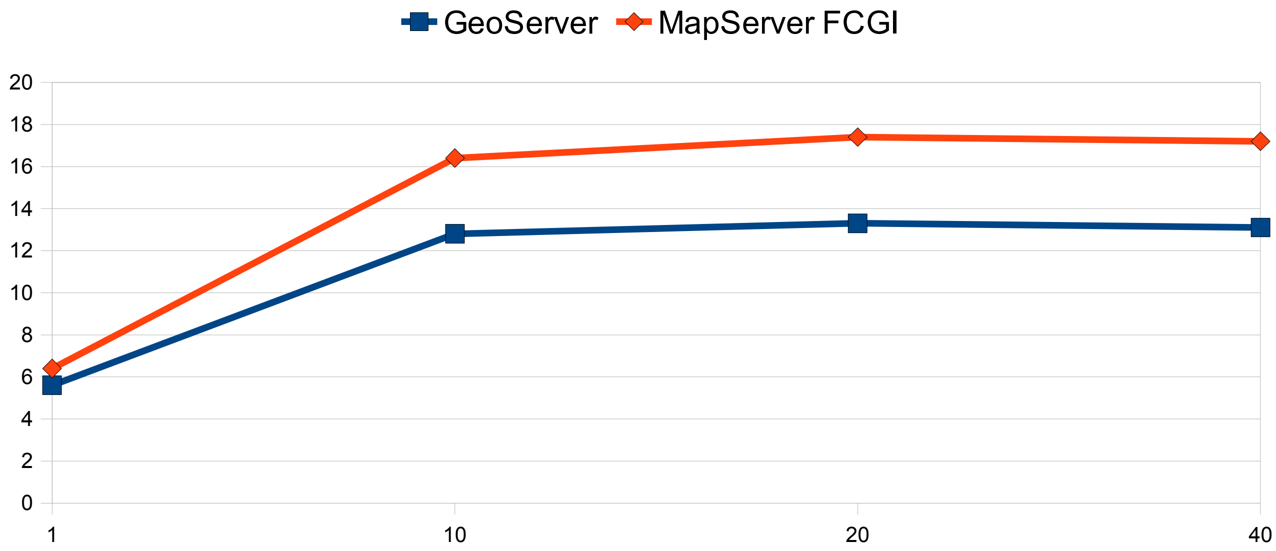
Label	GeoServer	MapServer FCGI
1	6.6	8
10	14.3	18.6
20	14.9	19.4
40	14.7	19.1

T4: Oracle



Label	GeoServer	MapServer FCGI
1	6	6.8
10	13.2	16.8
20	13.9	17.5
40	13.5	17.3

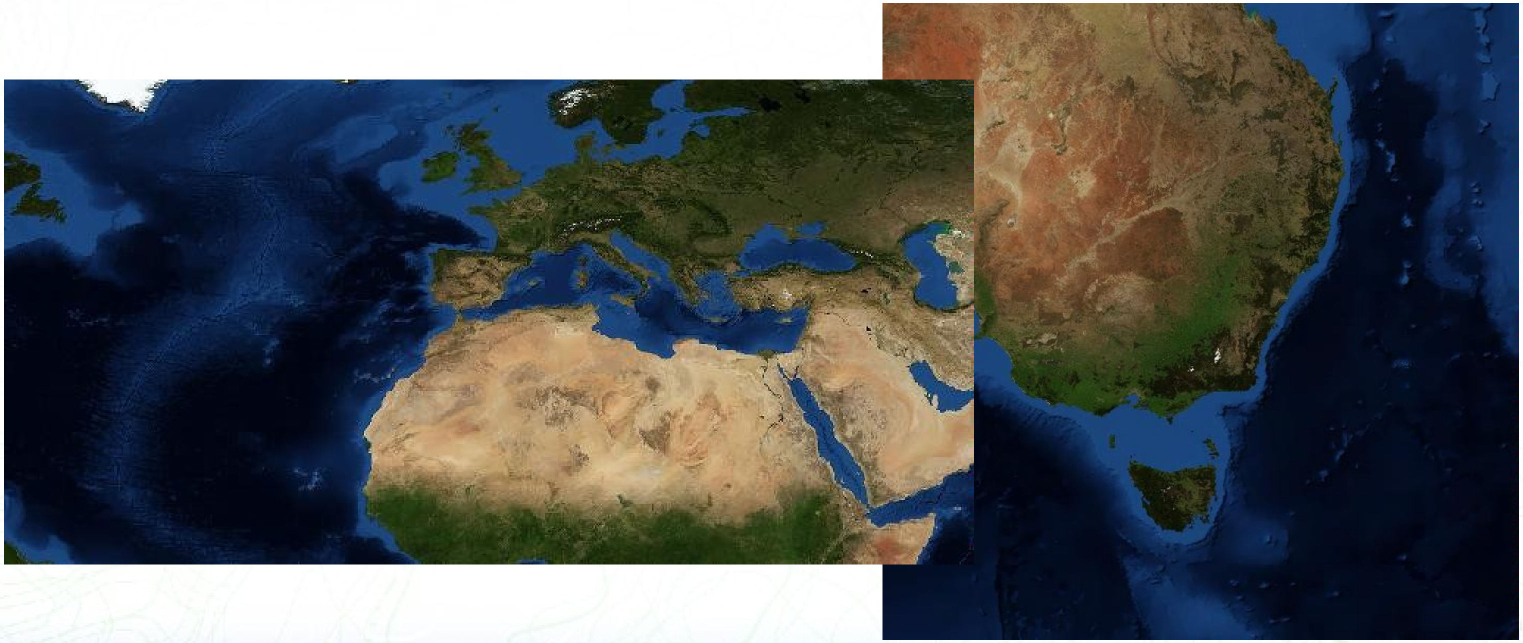
T4: SDE



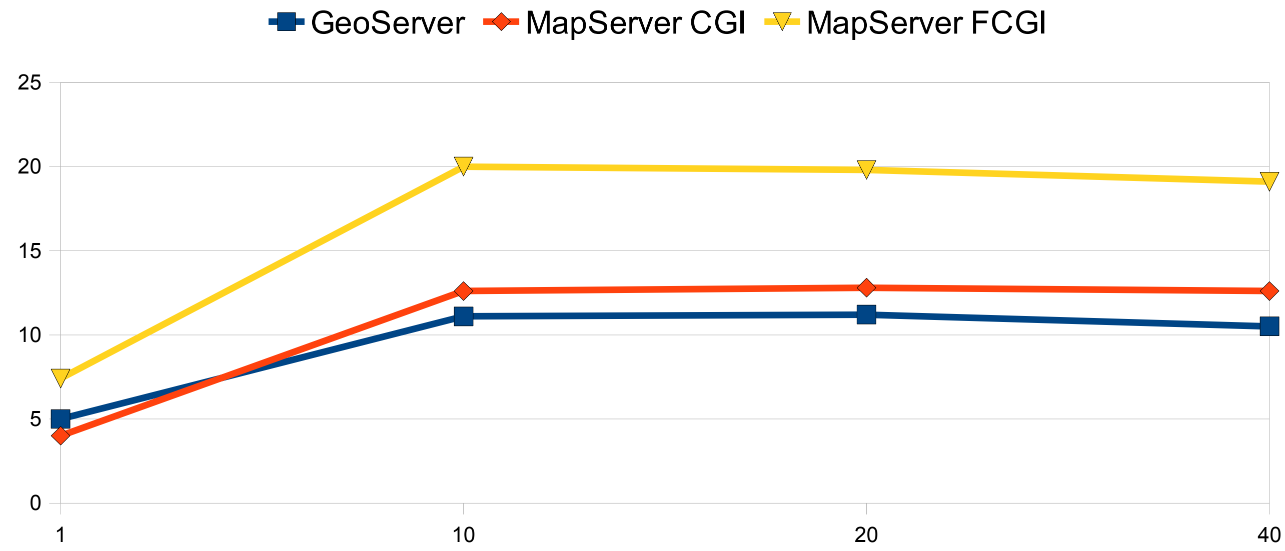
Label	GeoServer	MapServer FCGI
1	5.6	6.4
10	12.8	16.4
20	13.3	17.4
40	13.1	17.2

T5: raster

- Bluemarble TNG, as single ECW file, 512 mosaic TIFF, and BigTIFF (MapServer) vs 8 tiles mosaic TIFF (GeoServer)
- ECW file is 260MB
- TIFFs are uncompressed, tiled, with overviews. Around 16GB for each mosaic

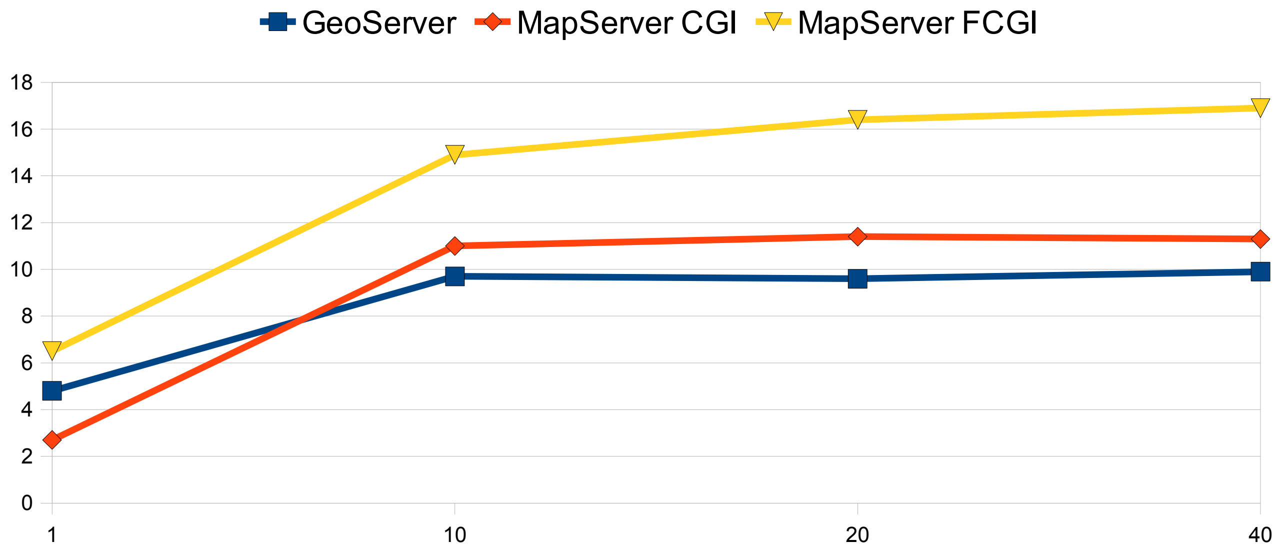


T5: ECW



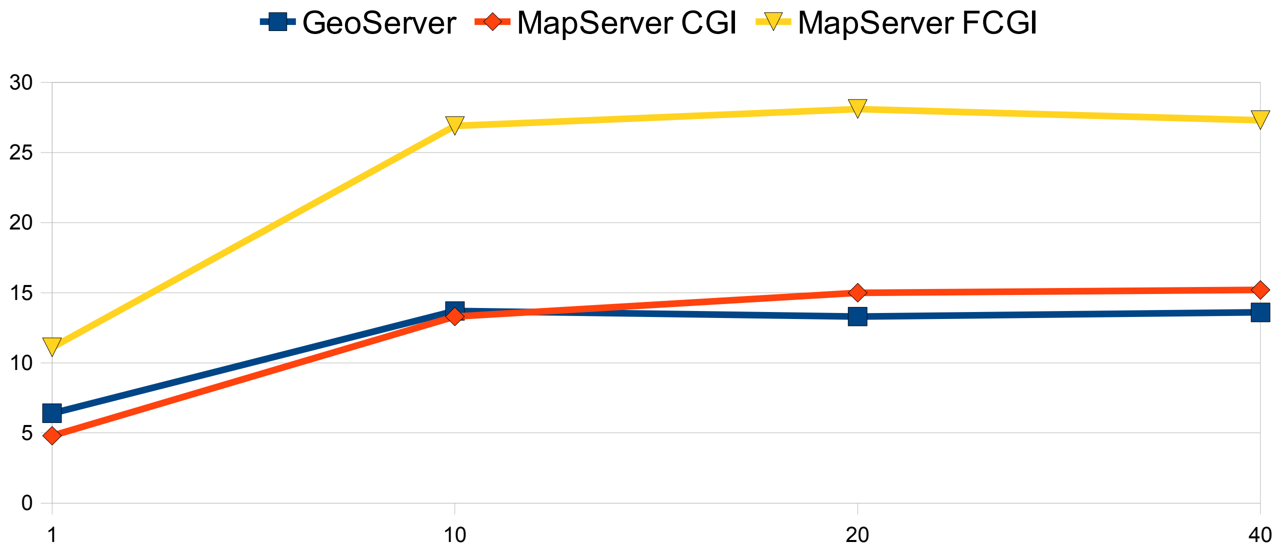
Threads	GeoServer	MapServer CGI	MapServer FCGI
1	5	4	7.4
10	11.1	12.6	20
20	11.2	12.8	19.8
40	10.5	12.6	19.1

T5: 512 tiles mosaic



Threads	GeoServer	MapServer CGI	MapServer FCGI
1	4.8	2.7	6.5
10	9.7	11	14.9
20	9.6	11.4	16.4
40	9.9	11.3	16.9

T5: BigTiff vs 8 tiles tiff



Threads	GeoServer	MapServer CGI	MapServer FCGI
1	6.4	4.8	11.1
10	13.7	13.3	26.9
20	13.3	15	28.1
40	13.6	15.2	27.3

GeoServer cannot do BigTiff, so a 8 tiles mosaic was used (almost 2G B per tile)

FOSS4G 2009 improvements

- Several code enhancements were made during this year's exercise:
  - MapServer
    - Fixed problem with labels following features
    - Fixed problem with 5.6.0-beta for SDE layers
    - Found problem with 5.6.0-beta for Oracle fastcgi and memory leak
  - GeoServer
    - Fixed problem with Oracle connection pooling and index scanning
    - Fixed problem with ECW crashes under high load

Hopeful Changes for Next Shootout

- Involvement of more WMS servers
- Put the servers under a faster internet connection for quicker data transfers and tests setups
- Modify the number of parallel client requests performed so that the 2-4-8 parallel clients case gets exposed
- Phases/cycles for benchmarking – to allow for teams to interpret results and make software enhancements
- Reprojection in both vector and raster data
- Audience suggestions?

More Information

- Scripts, configuration files, results stored in OSGeo SVN: http://svn.osgeo.org/osgeo/foss4g/benchmarking/
- Wiki home: http://wiki.osgeo.org/wiki/Benchmarking_2009
- Mailing list: http://lists.osgeo.org/mailman/listinfo/benchmarking