

# WMS Benchmarking

Cadcorp GeognoSIS, Contellation-SDI,  
ERDAS APOLLO, GeoServer, Mapnik,  
MapServer, Oracle MapViewer,  
QGIS MapServer



# Executive summary

- Compare the performance of WMS servers
  - 8 teams
- In a number of different workloads:
  - Vector: native (EPSG:4326) and projected (Google Mercator) street level
  - Raster: native (EPSG:25831) and projected (Google Mercator)
- Against different data backends:
  - Vector: shapefiles, PostGIS, Oracle Spatial
  - Raster: GeoTiff, ECW Raster





# Benchmarking History

- 4th FOSS4G benchmarking exercise. Past exercises included:
  - FOSS4G 2007: Refrations 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
  - FOSS4G 2009: MapServer and GeoServer teams in a cooperative benchmarking exercise
- Friendly competition: goal is to improve all software



# Datasets Used: Vector

Used a subset of BTN25, the official Spanish 1:25000 vector dataset

- 6465770 buildings (polygon)
- 2117012 contour lines
- 270069 motorways & roads (line)
- 668066 toponyms (point)
- Total: **18 GB** worth of shapefiles







# Datasets Used: Raster

Used a subset of PNOA images

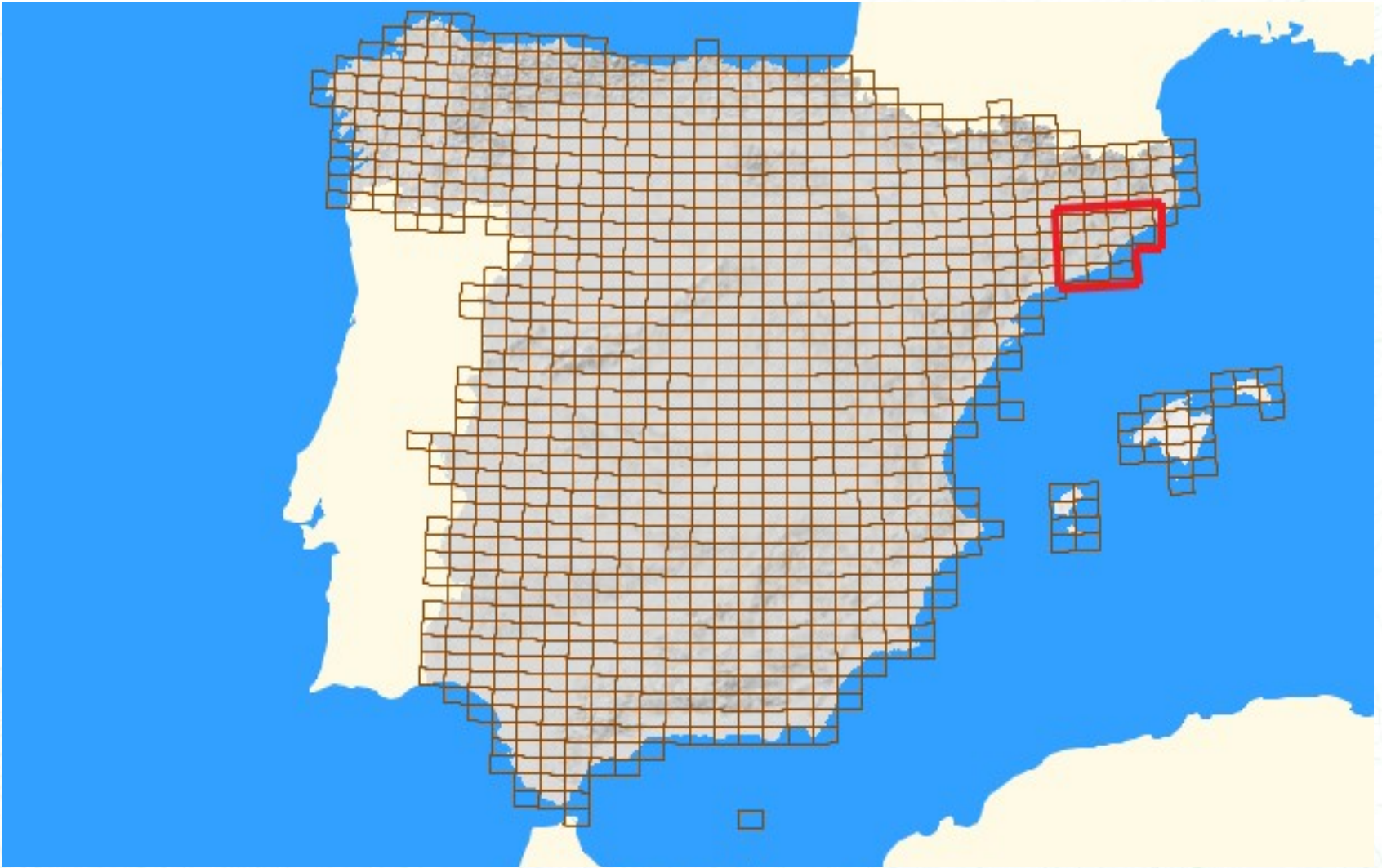
- 50cm/px aerial imagery, taken in 2008
- 56 GeoTIFFs, around Barcelona
- Total: 120 **GB**





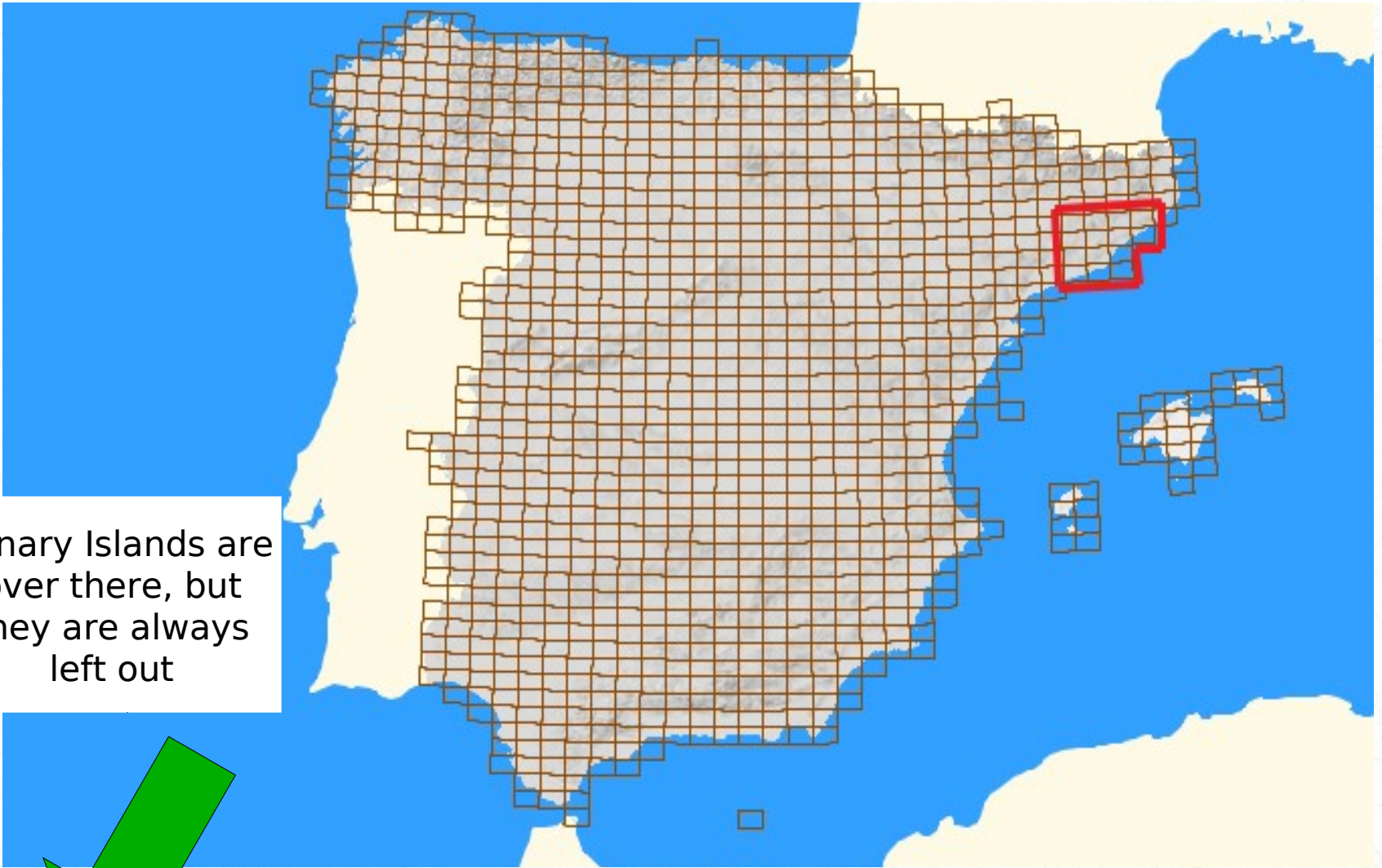


# Datasets Used: Extents





# Datasets Used: Extents



Canary Islands are over there, but they are always left out







# Datasets Used: Credits

Both BTN25 and PNOA are products of the **Instituto Geográfico Nacional**. Any non-commercial use of the data (such as benchmarks) is allowed.

You too can download the data used in the benchmark by visiting:

<http://centrodedescargas.cnig.es/CentroDescargas/>







# Datasets Used: Issues

- **Real data, real problems**
- .shp files bigger than 2 GB
  - Contours had to be split in 7 shapefiles
- .dbf files bigger than 2GB
  - Problems accessing the attributes of some features
  - Caused some servers to not filter features by attribute
- Non-ASCII characters (ó, ñ, ç) not in UTF-8
  - Some servers had problems rendering these characters



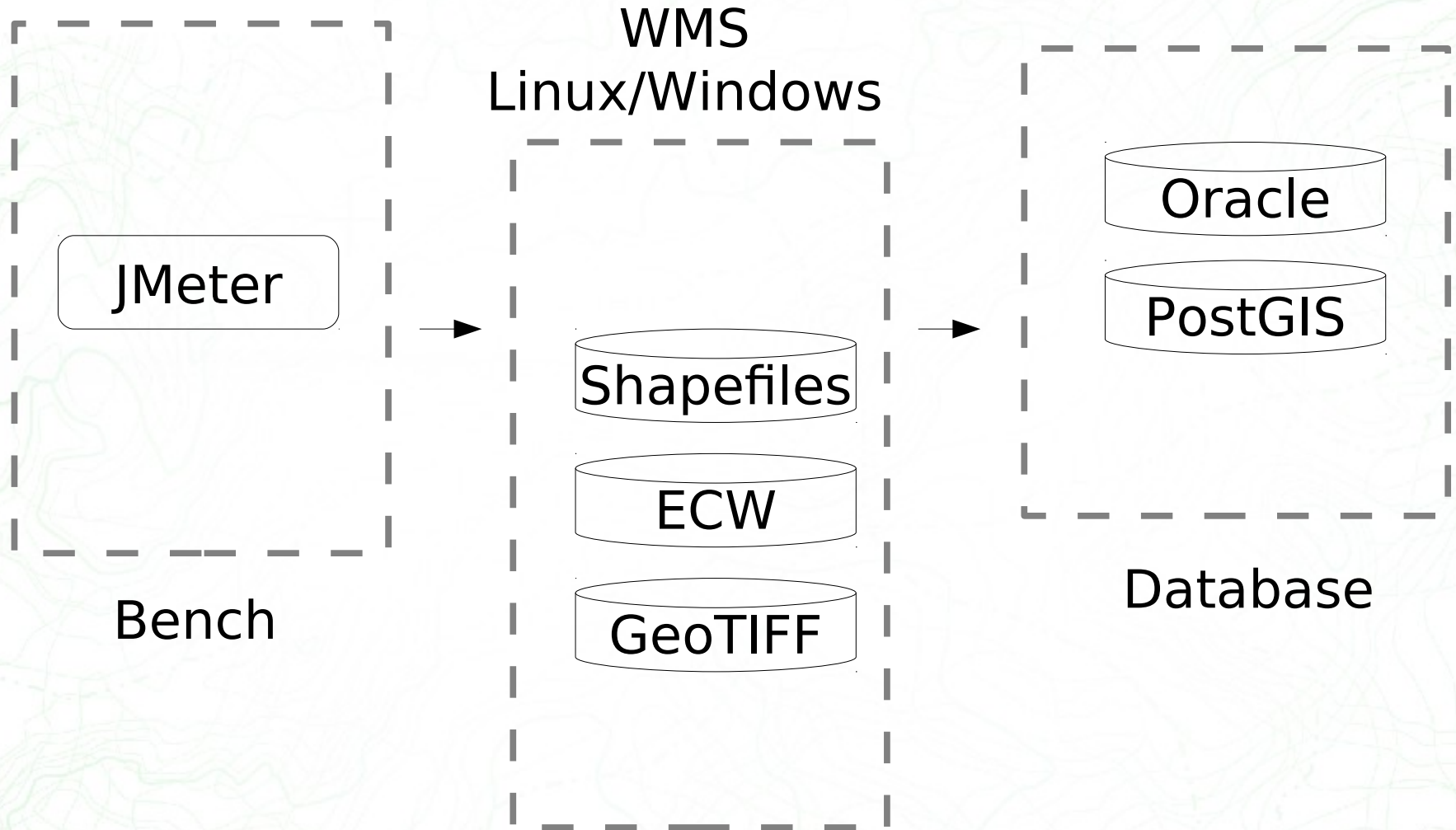
# Rules of engagement

- 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
  - Windows and Linux servers, 2 separate identical servers





# Hardware Configuration





# Hardware specs

- JMeter:

- Dell Precision Workstation 390 from 9/7/2006

- Processor, 6300, 1.86, 2M, Core Duo-conroe, Burn 2
  - 2Gb RAM 160 Gb Hard drive 7200 rpm OS: Centos 5.5 i386

- WMS(2):

- Dell PowerEdge R410 - Ship Date: 7/7/2010

- Processor: Intel® Xeon® E5630 2.53Ghz, 12M Cache, Turbo, HT, 1066MHz Max Mem
  - 8GB Memory (4x2GB)
  - 2TB 7.2K RPM SATA
  - OS: Windows Server 64bit, Centos 5.5 x86-64

- Database:

- Gateway E6610D Intel Core2 Duo - E6750 2.66 Ghz
  - 250Gb Hard Drive 7200 rpm, 4Gb Ram
  - OS: Centos 5.5 x86-64





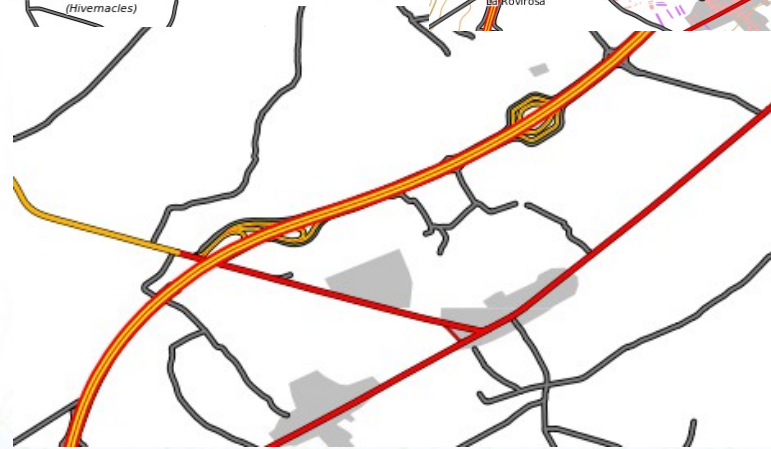
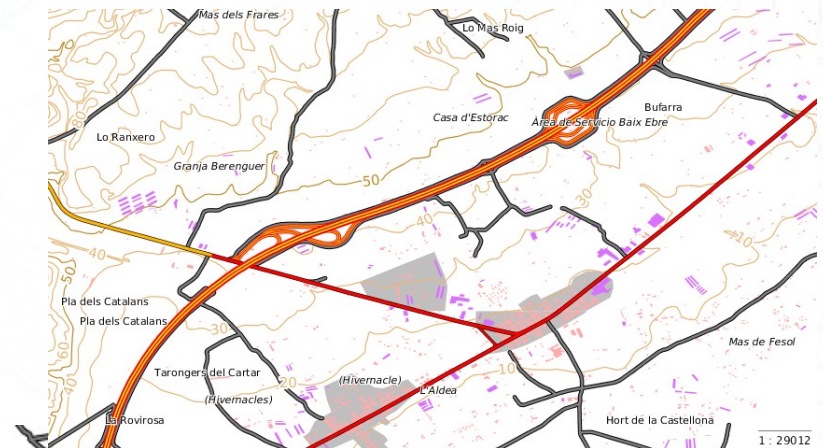
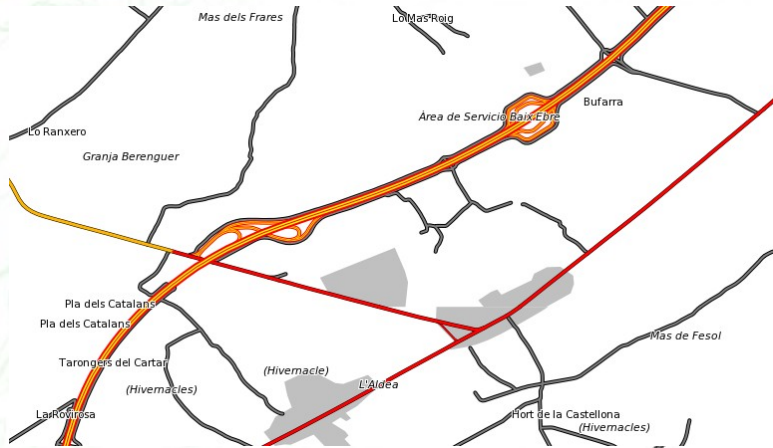
# Methodology

- Each test run performs requests with 1, 2, 4, 8, 16, 32 and 64 parallel clients (for a total of 2152 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



# Vector Output

- Vectors without projection (EPSG:4326), with (EPSG:3857)
- PNG24 output AntiAliased, Scale dependent rendering







# Raster Output

- Rasters without projection (EPSG:25831) with (EPSG:3857)
- JPEG output (90% quality)





# Resource consumption notes

- During a benchmark usually a single resource is used to its limits and plays the bottleneck
- Common bottlenecks are the CPU, the disk access, the network access, the remote database access
- Looking at system monitoring on the WMS machines during the benchmark runs, the disk was playing the role of the bottleneck in most tests. For some servers, for some runs, the data became fully cached and performance increased dramatically





# Disk-bound v.s. unbound scenarios

- The initial bottleneck in the benchmark tests was the disk read access
- Some teams managed to turn this *disk-bound* scenario into a *disk-unbound* scenario, allowing servers to fully run in memory, with no disc access and putting all load on the CPU
- Results observed in the *disk-unbound* scenario were 5 to 10 times better than in the *disk-bound* scenario
- This was possible thanks to the conjunction of the following:
  - the Red Hat Enterprise Linux sever is efficiently caching data blocks at the OS level
  - the 2000 WMS requests are the exact same between all runs
  - the Server RAM memory is of 8GB



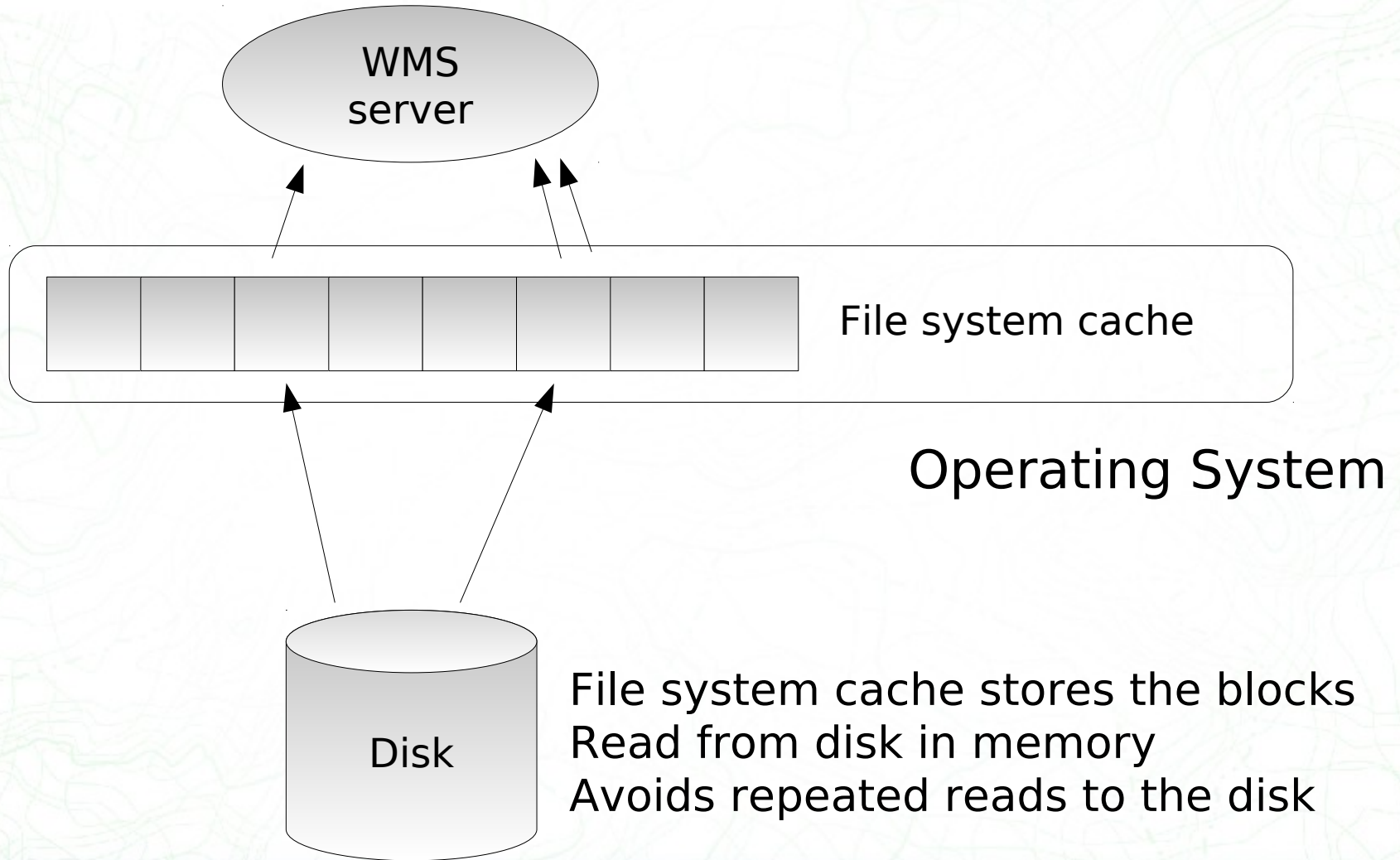
# Disk-bound v.s. unbound scenarios

- Unfortunately, the Windows server did not behave the same way as RHEL
- 
- The Windows server was down in the last benchmark days, preventing some teams to try to reproduce the disk unbound scenario
- 
- As apples cannot be compared to oranges, all participants did agree that **teams were not mandatory to publish their benchmark results in the final presentation**



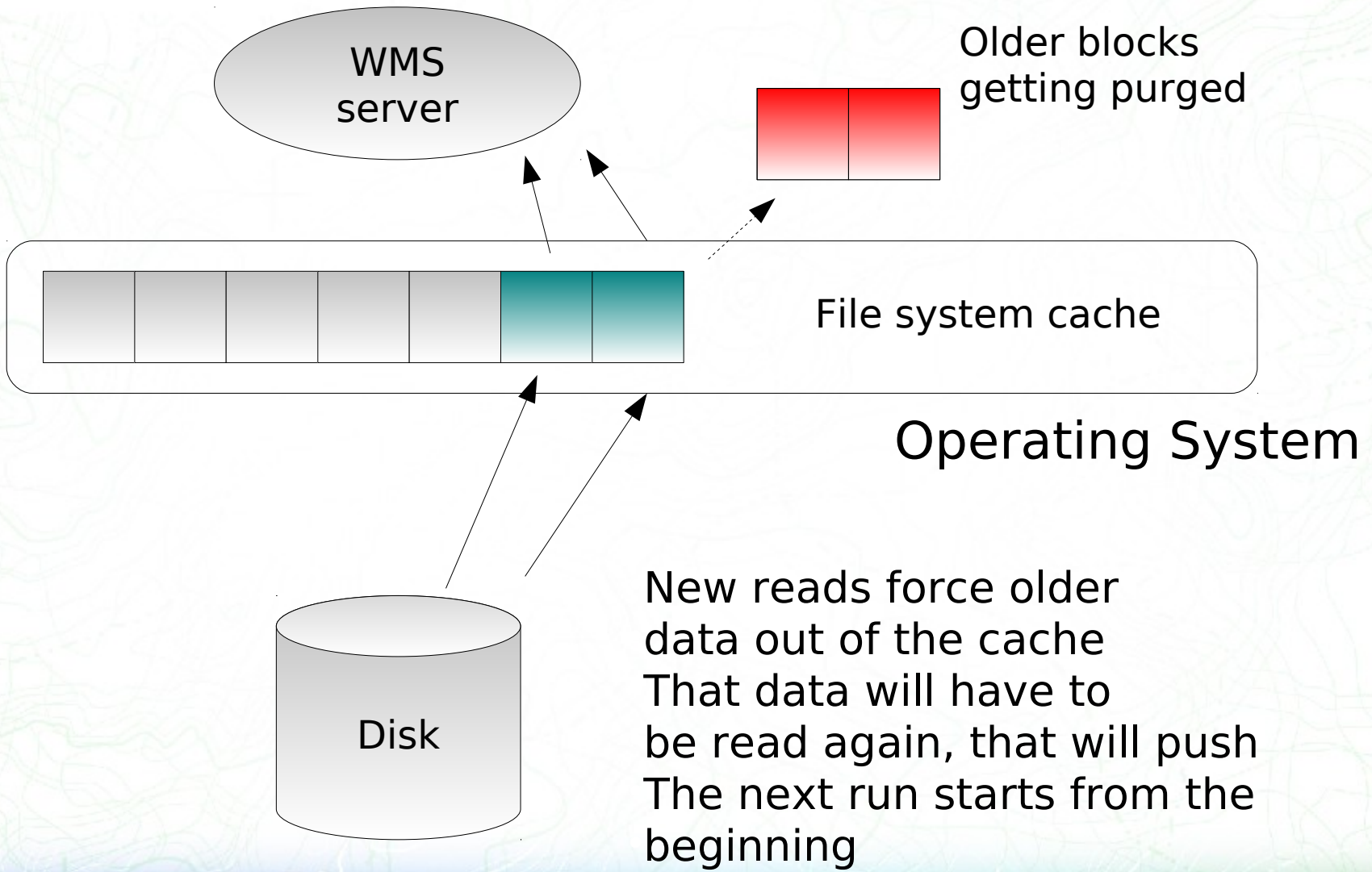


# I/O reading schematics





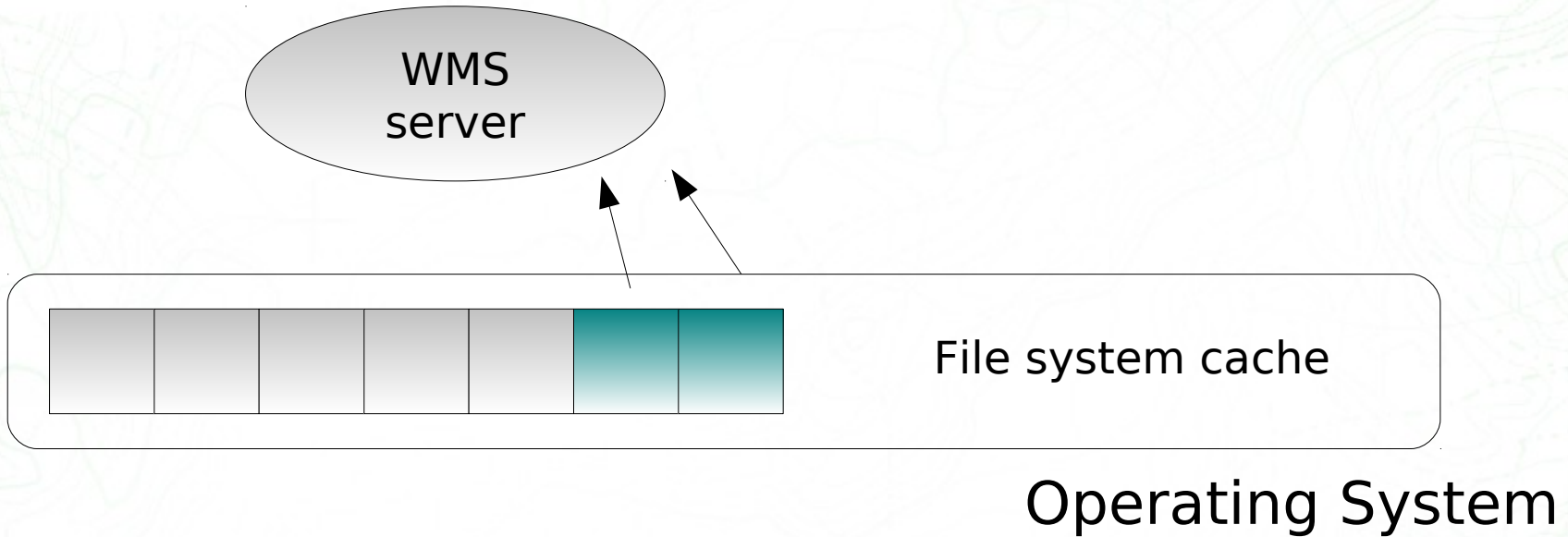
# Disk bound situation







# CPU bound situation





# Server/Team Findings



La Rambla, Barcelona, 2010-09-07





# GeoServer: Overview

[www.geoserver.org](http://www.geoserver.org)

- Versions tested
  - GeoServer 2.1 daily build (similar to 2.1 Beta, but not equal)
- Individuals involved
  - Andrea Aime: setup, vector optimizations, raster optimizations
  - Simone Giannecchini, Daniele Romagnoli: raster optimizations
  - Ivan Grcic: extensive raster testing
  - Justin Deoliveira and Jody Garnett: suggestions and more vector tests



# GeoServer: Successes

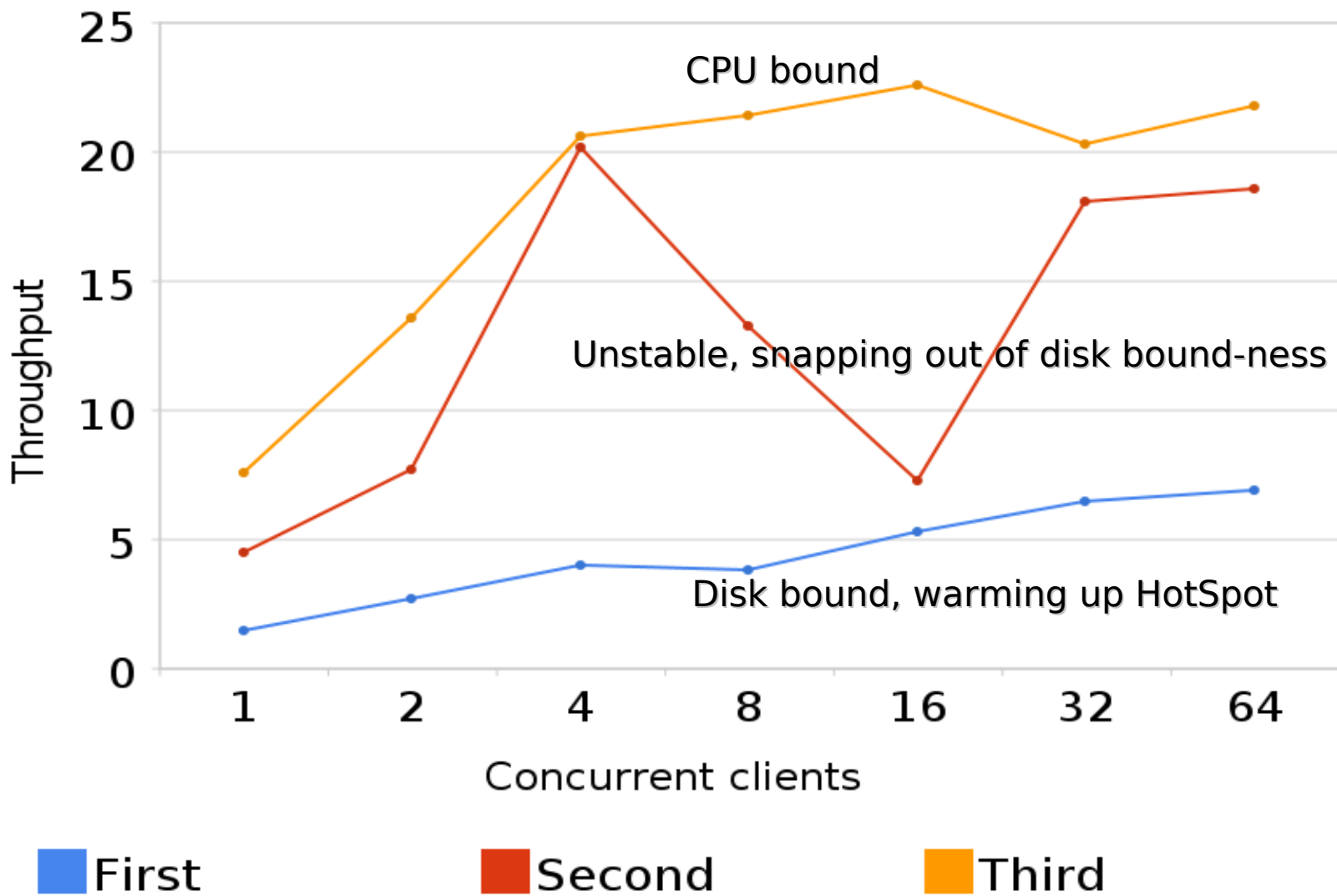
- Software improvements
  - Reading DBF files larger than 2GB
  - Large dataset/multilayer rendering speedups
  - Fast rectangular clipping of large geometries (clip before render)
  - Faster pure raster rendering path
  - General improvements in the shapefile reader to open less files and read less from disk
  - Improving spatial indexing to get compact yet selective indexes





# GeoServer: Challenges

- Disk vs CPU bound
  - Lot of work went into optimizing I/O assuming a disk bound test
  - The test easily flips from being disk bound to being CPU bound
  - At some point the optimizations made it possible for all data to sit in the operating system cache
- Raster reprojection
  - Other teams use local approximation techniques to speed up transformation (use of local linear transform)
  - We did not make it in time to implement it this year, but it's in the pipeline

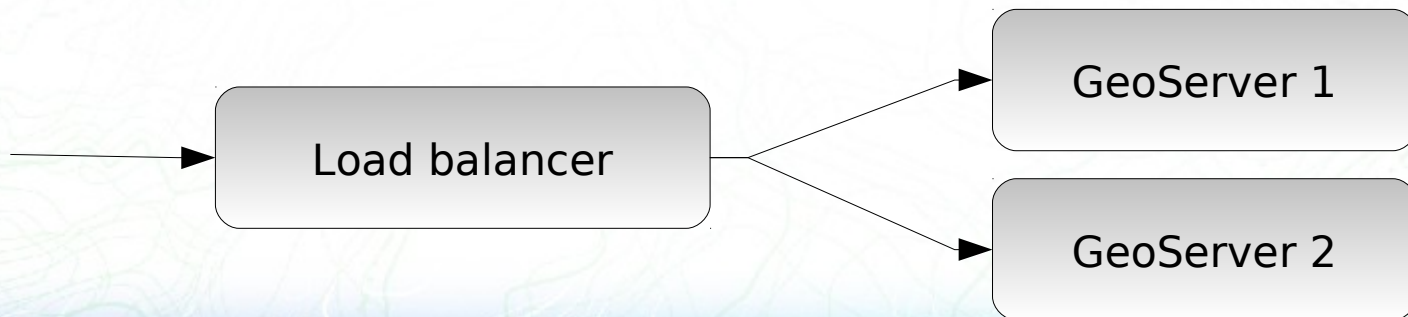






# Java servers Achille's heel

- Sun Java2D can rasterize only one shape at a time in the whole process when antialiasing is enabled
- OpenJDK can rasterize multiple in parallel, but it's very slow
- [http://bugs.sun.com/view\\_bug.do?bug\\_id=6508591](http://bugs.sun.com/view_bug.do?bug_id=6508591)
- Affects APOLLO, Constellation, GeoServer, MapViewer
- Suggested setup: when you have more than 4 CPU, and if you are normally CPU bound, setup multiple processes and use a load balancer to distribute the load among them





# MapServer: Overview

- Background
  - Tested on Linux and Windows (x64)
- Versions tested
  - MapServer 5.7-dev
  - Apache 2.2 with mod\_fcgid
  - GDAL 1.7-dev
- Individuals involved
  - Jeff, Mike, Daniel, Frank, Alan, Julien, SteveL

[www.mapserver.org](http://www.mapserver.org)



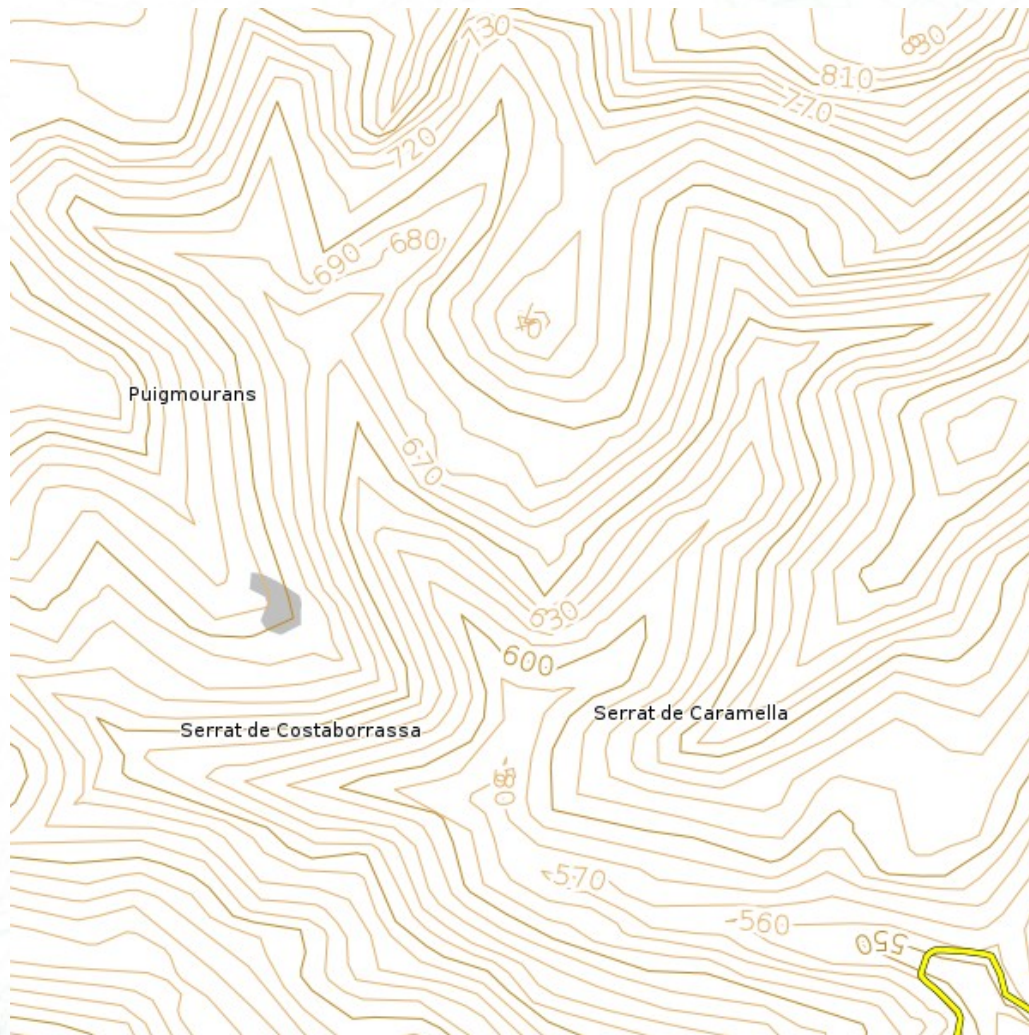


# MapServer: Improvements

- Previous years
  - improvements in large shapefile indexing
  - raster read optimization (single pass for multiple bands)
  - enhancing polygon labelling placement
  - EPSG codes caching
  - PostGIS improvements
  - Oracle improvements
- Current
  - large DBF support (> 2GB)
  - improving labels on curved lines
  - Improved label formatting
  - discovered raster reprojection sampling parameter



# Labelling Contours (1)







# Labelling Contours (1)



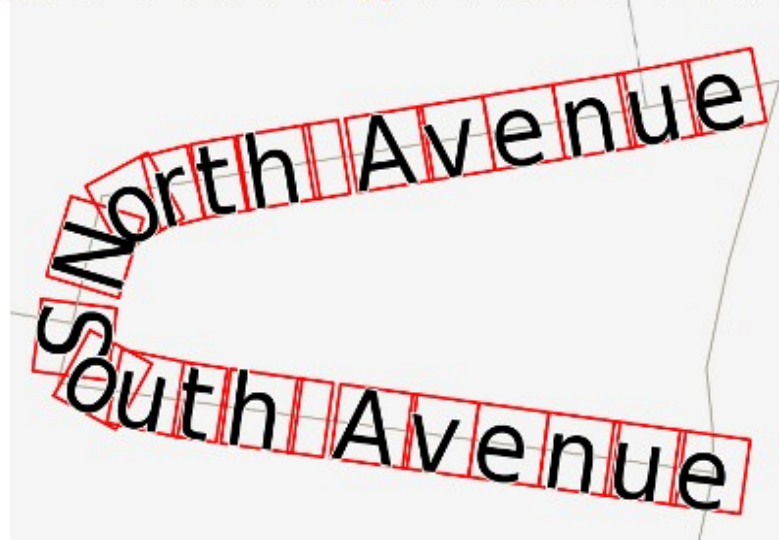


# MapServer: Label Overlap Angle

MAXOVERLAPANGLE 0



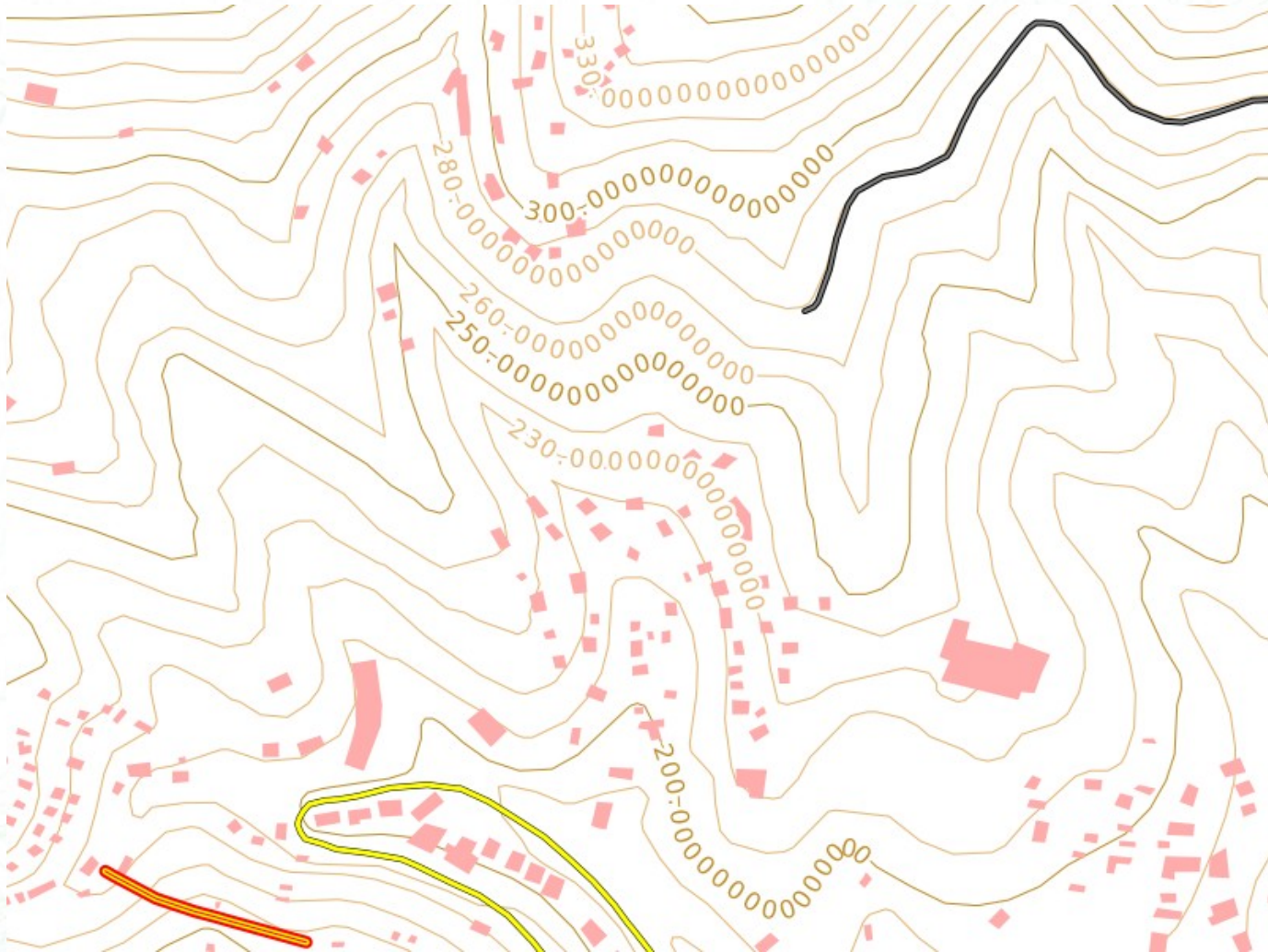
MAXOVERLAPANGLE 22.5







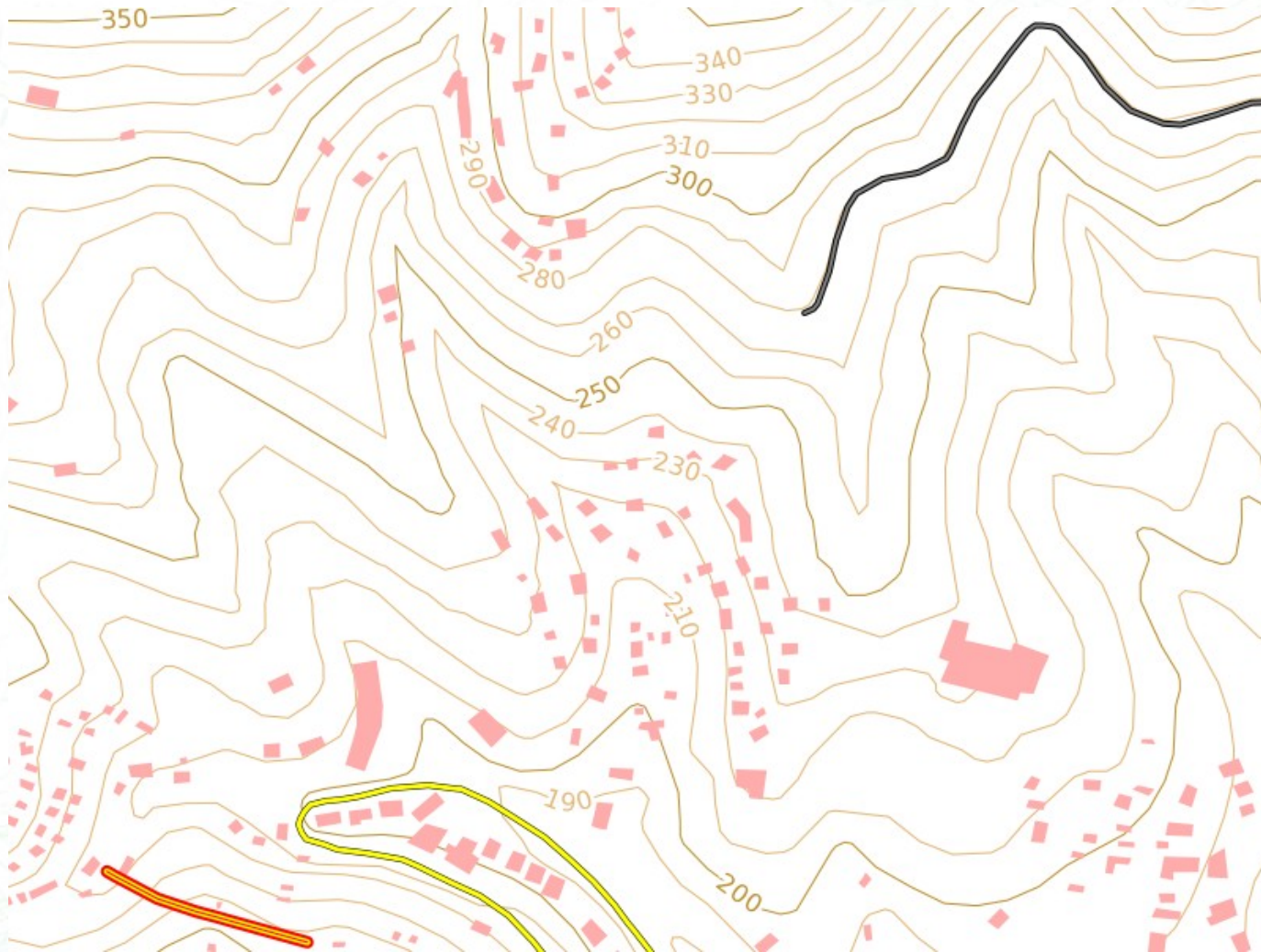
# Labelling Contours (2)







# Labelling Contours (2)







# MapServer – Raster Resampling

- 2x oversampling - ~ 2.5 requests/sec
- MapServer Default







# MapServer – Raster Resampling

- Added PROCESSING "OVERSAMPLE\_RATIO=1"
- Result: 50 requests/sec – only due to disk bound problem







# MapServer: Challenges

- Not enough time in the day to run all tests
- Debate: speed optimization vs map image output quality
- No time spent on a “best run”, although a lot of optimization was performed for “base run”
- Having been involved in previous years, able to complete more tests than some newcomers, but we were still incomplete



# Mapnik: Overview

- Background [www.mapnik.org](http://www.mapnik.org)
  - Tested on Linux (windows next year)
  - Written in C++, boost, agg, cairo
- Versions tested
  - Rendering library: Mapnik trunk (aka Mapnik2)
  - Server: Paleoserver 0.1 (boost::asio multithreaded http)
  - Server2: mod\_mapnik\_wms (apache module)
- Individuals involved
  - Matt Kenny ([mkgeomatics.com/](http://mkgeomatics.com/)) designed styles
  - Artem Pavlenko and Robert Coup - features and fixes





# Mapnik: Successes

- Lessons:
  - First year – awesome, humbling, excited for next
  - Collaboration takes commitment and positive attitude
- Software improvements
  - New C++ async, threaded WMS server (paleoserver)
  - More efficient shapefile reading
  - Caching of features when >1 style applied to layer
- Benefits for team, for users, for community...
  - Paleoserver scales excellently, light threading model ++
  - Mapnik + PostGIS + 32 threads (Cores \* 4) = FAST



# Mapnik: Challenges

- Tiles, PostGIS, OSM
- Too many threads (Cores\*2) with shapefiles == disk bound
- Rasters and vector reprojection need attention





# Oracle MapViewer: Overview

- **Background** [oracle.com/technetwork/middleware/mapviewer](http://oracle.com/technetwork/middleware/mapviewer)
  - Oracle MapViewer is a J2EE server component, and a part of Oracle's Fusion Middleware offering
  - Tests were done on Linux, connecting to Oracle 11g R2
  - MapViewer can run on all J2EE compatible containers
- **Versions tested**
  - MapViewer 11g R2 development builds
- **Individuals involved**
  - LJ Qian, Dan Geringer, and a big "Thank you" to Mike Smith who setup the initial MapViewer/Oracle DB and did all the map layer styling.



# Oracle MapViewer: Successes

- Lessons:
  - Overall we found this event extremely helpful
- Software improvements
  - Fixed a CASED line bug where features were merging cross themes.
  - Added generic middle-tier CS transformation and user configurable DPI
  - Completed shapefile support (but did not have time to test it).
  - Identified some major bottlenecks in MapViewer





# Oracle MapViewer: Challenges

- Problems or difficulties encountered
  - How to better utilize large amount of memory in the middle tier (e.g., caching)
  - How to make use of GDAL and OGR tools
  - Plan to continue use of benchmark servers for further testing
  - Plan to participate next year
  - No raster results, db server not equivalent to file servers in performance, not fair comparison. Will have equivalent db server next year



# Oracle MapViewer: Results

- We average 17 requests /s for both 4326 and 3857 vector tests, connecting to Oracle Spatial via JDBC thin driver.
- MapViewer was running with JDK 6, with 1.2GB of heap memory.
- A database connection pool of 16 was used by MapViewer
- There is definitely room for improvements!





# QGIS Mapserver: Overview

[www.qgis.org](http://www.qgis.org)

- FCGI WMS server for publishing QGIS projects
- Focused on integration with desktop application
- QGIS desktop and WMS server use same rendering library
  - Uses libqgis\_core / QImage for map symbolisation and rendering
  - Desktop users benefit from all improvements in QGIS mapserver
  - Server image looks the same as in desktop GIS (exception: image compression)
- WMS server will be included in QGIS 1.6
- QGIS Mapserver is on the OSGeo live DVD



# QGIS Mapserver: Successes

- Versions
  - Threading branch (gsoc)
  - SVN trunk
  - Symbology-ng (except for contours)
- Software improvements during benchmarks
  - Fixed some memory leaks
  - Adapted scale calculation to be comparable with other servers
  - Symbol levels in rule based renderer





# QGIS Mapserver: Challenges

- Benchmark problems
  - Fallback to svn trunk version
  - Problems with the contour layer ( probably related to geometry clipping )
  - Not enough time for getting consistent results
- Work to be done after the conference:
  - Testing each layer individually to find bottlenecks
  - More examination on contour layer, with/without labeling
  - See what can be improved on QGIS mapserver / QGIS level and what needs work on render engine level (Qt libraries)



# Constellation-SDI

[www.constellation-sdi.org](http://www.constellation-sdi.org)

- Background
  - OGC web services platform: CSW, SOS, WMS, WMTS, WCS...
  - JEE server deployed on gnu/linux and windows
  - Core based on **Geotoolkit**, web apps based on **MapFaces**
  - Free software (LGPL)
- Versions tested
  - Improvements will land in version 0.7, after formal review
- Individuals involved
  - Cedric Briançon, Martin Desruisseaux, Johann Sorel





# Constellation-SDI: Successes

- Lessons:
  - First serious work on performance *...not the last*
- Software improvements
  - Benchmarking: building test designs and tools
  - GeoTiff reader, TileManagers for unstructured mosaics
  - Raster performance: direct readers, fast reprojection
  - Vector performance: clean pipeline, file traversal issues
- Benefits
  - focus on profiling, load testing, stressing, vm/os envt



# Constellation-SDI: Challenges

- Good testing is *hard*
    - WMS is a portrayal service.
      - WMS output is not identical, cannot be basis of test
      - Output image size and compression vary
      - Raster images vary in quality
      - Vector plots vary in generalization, symbology
- => Tried to harmonize styles, forgot to check image sizes and compression levels, avoided discussions of quality although it is key to some participants





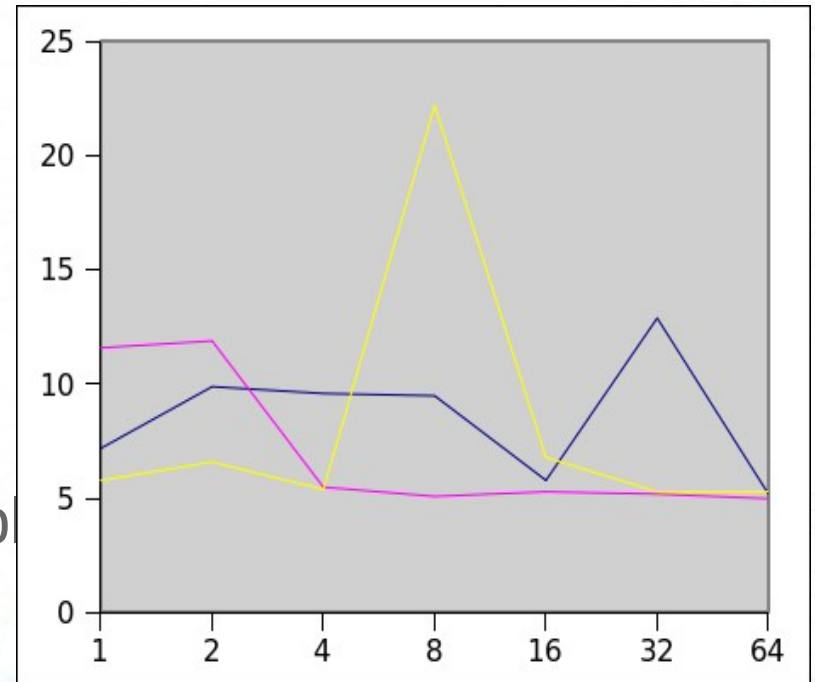
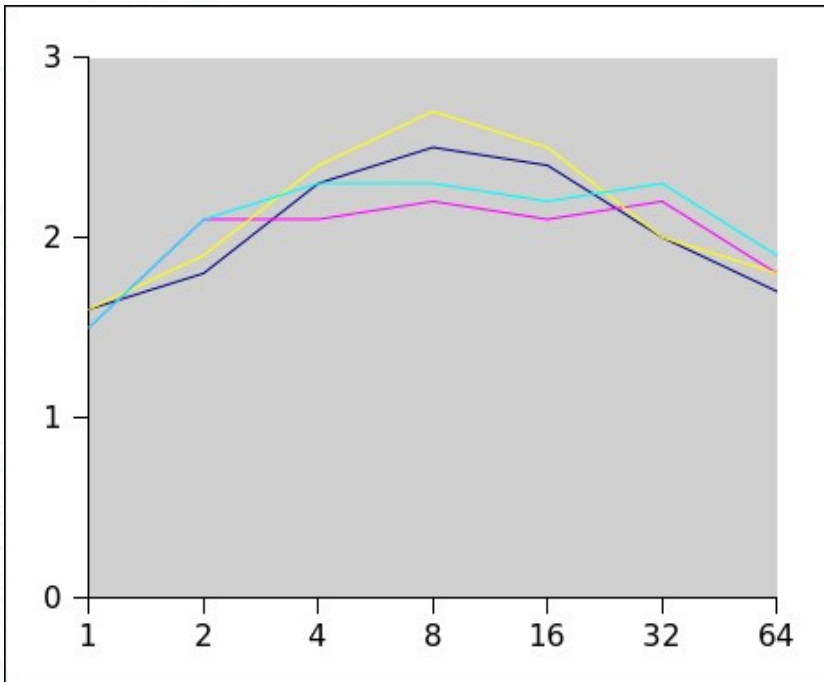
# Constellation-SDI: Challenges

- Good testing is *hard*
    - WMS is a portrayal service
    - WMS servers designed for many scales and uses
      - serve small datasets or national data repositories
      - use existing (ancillary) data or build custom data
- => Tests will be different to address performance of these different environment



# Constellation-SDI: Challenges

- Good testing is *hard*
  - WMS is a portrayal service
  - WMS servers designed for many scales
  - Testing metrics must satisfy several criteria



multip





# Constellation-SDI: Challenges

- Good testing is *hard*
  - WMS is a portrayal service
  - WMS servers designed for many scales
  - Testing metrics must satisfy several criteria
  - Testing takes time, addressing issues takes work

The 2010 testing regime was imprecise and insufficient, servers were down in the last critical hours, community organization unclear.

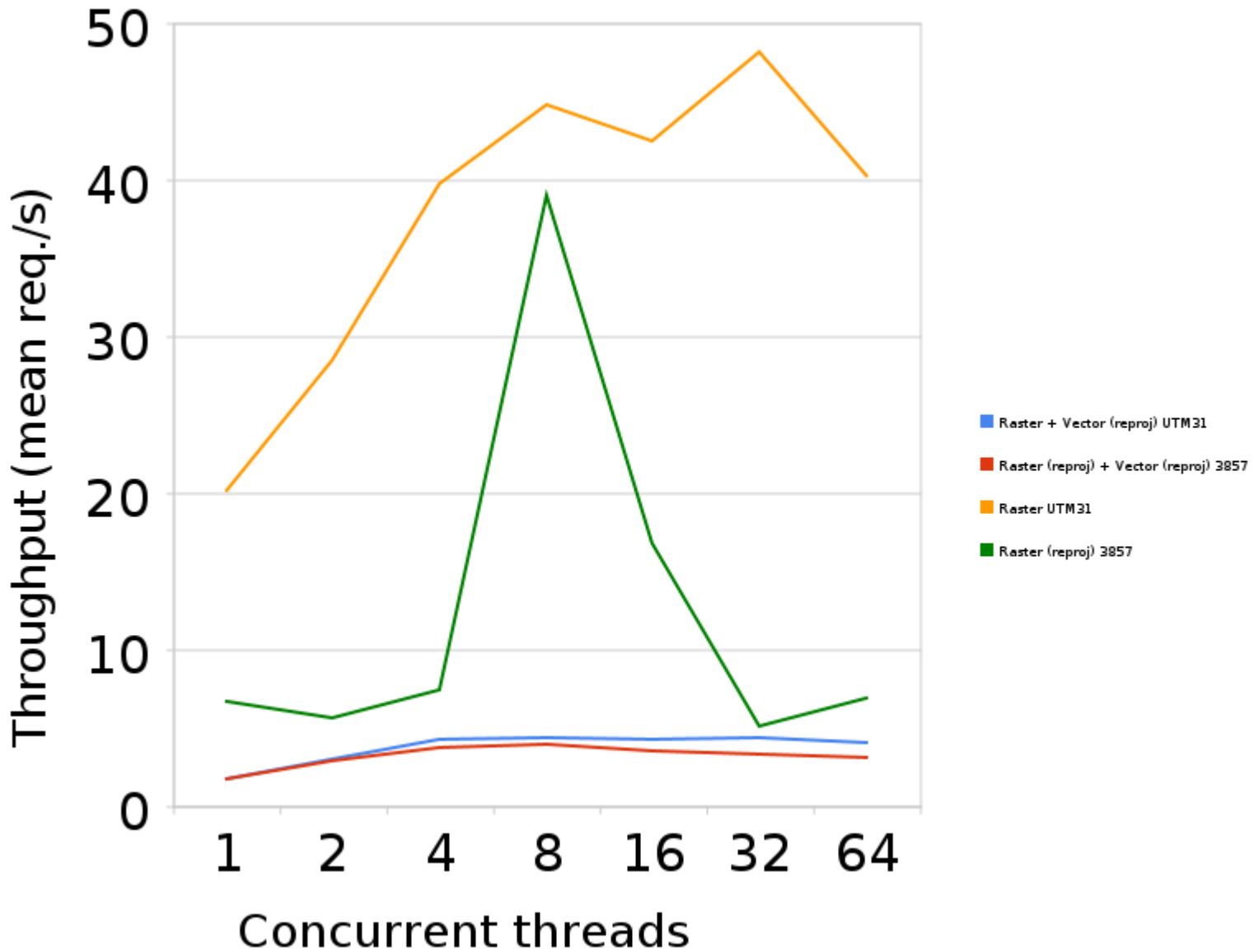
*=> some teams did not obtain final results*

*=> mean as metric not great, results unstable*

*=> next time around we hope to do better*



# Constellation-SDI: Results







# Cadcorp GeognoSIS: Overview

[www.cadcorp.com](http://www.cadcorp.com)

- Background
  - Entered to compete and to learn
  - Tested on Windows
  - C++
- Versions tested
  - 7.0 (current production release)
- Individuals involved
  - Two Martins: Daly and Schäfer



# Cadcorp GeognoSIS: Successes

- Lessons:
  - Rock solid stability up to 256 threads (in some unofficial tests)
- Software improvements
  - Handle DBF files > 2Gb; Support PostGIS geometry columns with 4D (XYZM) geometry type; Improved DBF string encoding handling; Cache DBF record data for the current SHP record; Allow Label Theme to use fill brush and/or outline pen from other Overlay Themes; Improved Label Theme "along lines" option, to better suit contour data
- Benefits for team, for users, for community...
  - Programmers using the product suite for a "real" project led to a few UI/UX improvements in desktop SIS
    - Miscellaneous bugfixes and improvements, above





# Cadcorp GeognoSIS: Challenges

- Our raster layer (above GDAL) is not as efficient as it could be
  - mloskot – behind you! - has already rewritten it since joining, for the next version of SIS: 7.1
- Very large shapefiles is not a typical use case for us
- We could not handle DBF files > 2Gb, so we fixed it
- Our Label Theme was poor for contour lines and only supported homogeneous styling, so we fixed those too



# Cadcorp GeognoSIS: Results

- Getting a kicking, as predicted ☺
- We should just use SHP .QIX files (viz shptree) and be done with it
- We've got plenty of work to do
- You can help!
  - <http://blog.lostinspatial.com/2010/09/07/cadcorp-is-recruiting-2/>
- We *will* be careful not to target this one (shapefiles and TIF mosaic) scenario





# ERDAS : Overview

- Background
  - First participation
  - Tested on Windows server
  - Java based for vector and native code for raster
- Versions tested
  - ERDAS APOLLO Essentials SDI & IWS 10.1
- Individuals involved
  - Anne-Sophie Collignon for the configurations, issues follow-up and tests runs
  - Liège and Perth APOLLO development teams for software improvements
  - Luc Donea for the overall follow-up



# ERDAS : Successes

- Lessons & benefits:
  - Exciting project, good discussion/collaboration between teams
  - Allow to concentrate on the software performance and rendering quality improvements.
  - Allow to experiment different server setup and document the preferred configuration for this kind of use-case
- Software improvements
  - For APOLLO IWS :
    - Upgrade to GDAL 1.7.2
    - Direct access to GDAL for Geotiff reader
    - Improved the mosaicing of TIF datasets to remove seam lines
  - For APOLLO SDI :
    - Tomcat 6 and Java 6: improved performances over Tomcat 5.5 and Java 5
  - Rendering :
    - Clash management for contour labels
    - Multipath rendering optimisation
    - Low level code optimisation
    - New option for shape index in memory





# ERDAS: Challenges

- Problems or difficulties encountered
  - Low resources : product release period coupled with holidays
  - First participation in FOSS4G WMS Shootout
  - Huge disk read bottleneck reducing rendering optimizations effects on the overall performances
  - Tests started very late because of data availability. As a result, issues were discovered late in the test design leaving no time to fix.



# ERDAS: Disk bound vs. CPU bound

- Some teams do experiment disk bounds scenarios while some others are running in a CPU bounds scenario
- ERDAS did experiment disk bound scenario on windows and was unable to investigate a CPU bound scenario : not enough time and Windows server down the last days.
- A usual use-case would be to have different requests using random BBOX. In this case, the OS data block caching would be a lot less effective bringing the servers back to a disk bound scenario





# ERDAS: Disk bound vs. CPU bound

- The use-cases met by ERDAS usually involve datasets that are bigger than the machine memory (RAM), and random distribution of the request BBOX. That's why ERDAS believes that the CPU bound scenario is not representative of ERDAS customers needs.
- As apples cannot be compared to oranges, disk bound scenario cannot be compared to CPU bound scenario.



# ERDAS: Conclusion

- As the benchmark participants could not reach a consensus on the results validity, the participants agreed that teams were not mandatory to publish their benchmark results in the final presentation
- Given the inconsistencies in the tests conception that were discussed, ERDAS is concerned that the different throughput results between server applications might confuse the community and mislead the community.
- ERDAS plans to conduct a webinar in the future to review the methodology and results of the FOSS4G benchmark. ERDAS will also provide analysis and ideas for future improvements.



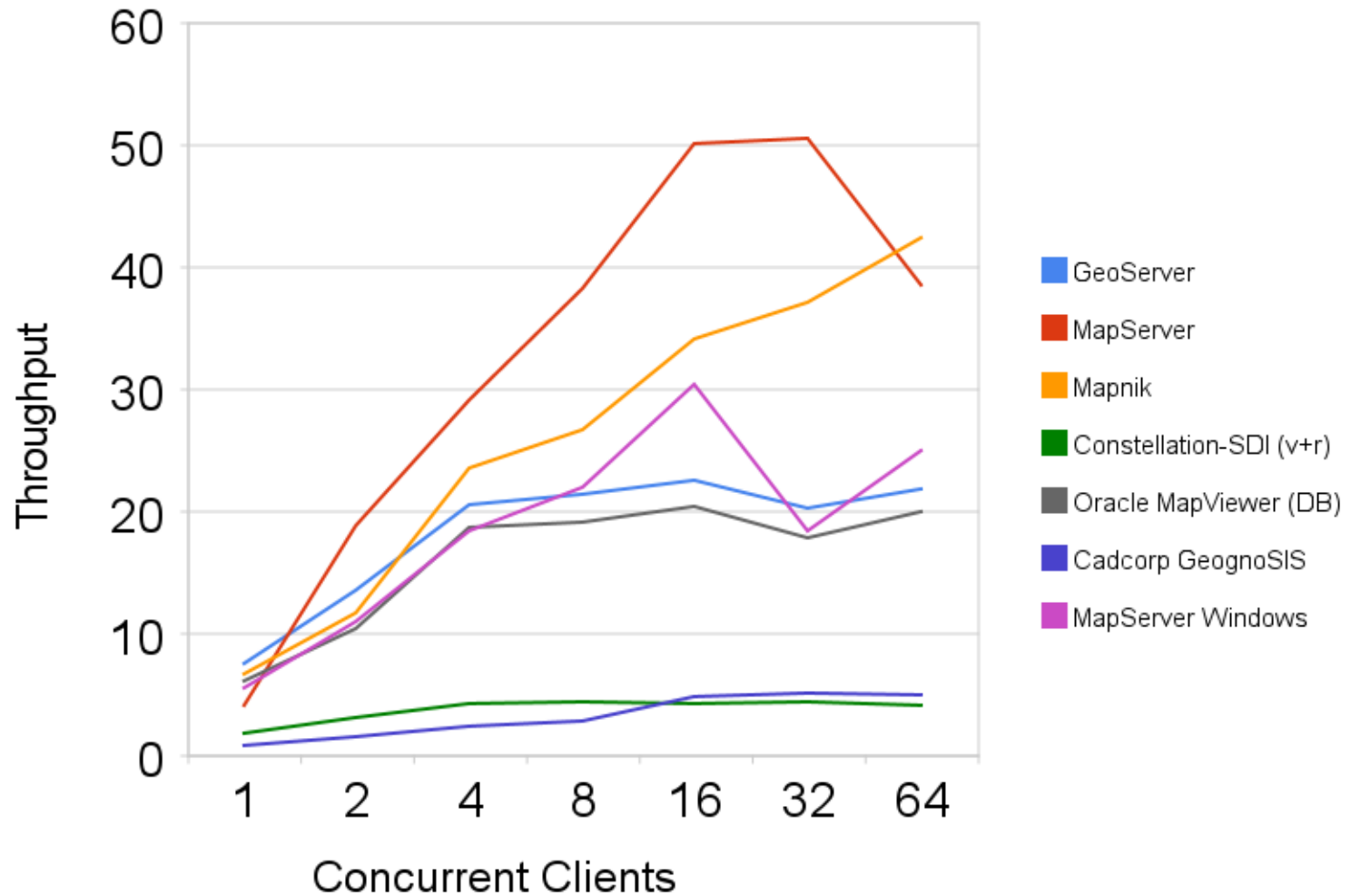


# Results



# Vector – No Reprojection (EPSG:4326)

Shapefile (no re-projection)

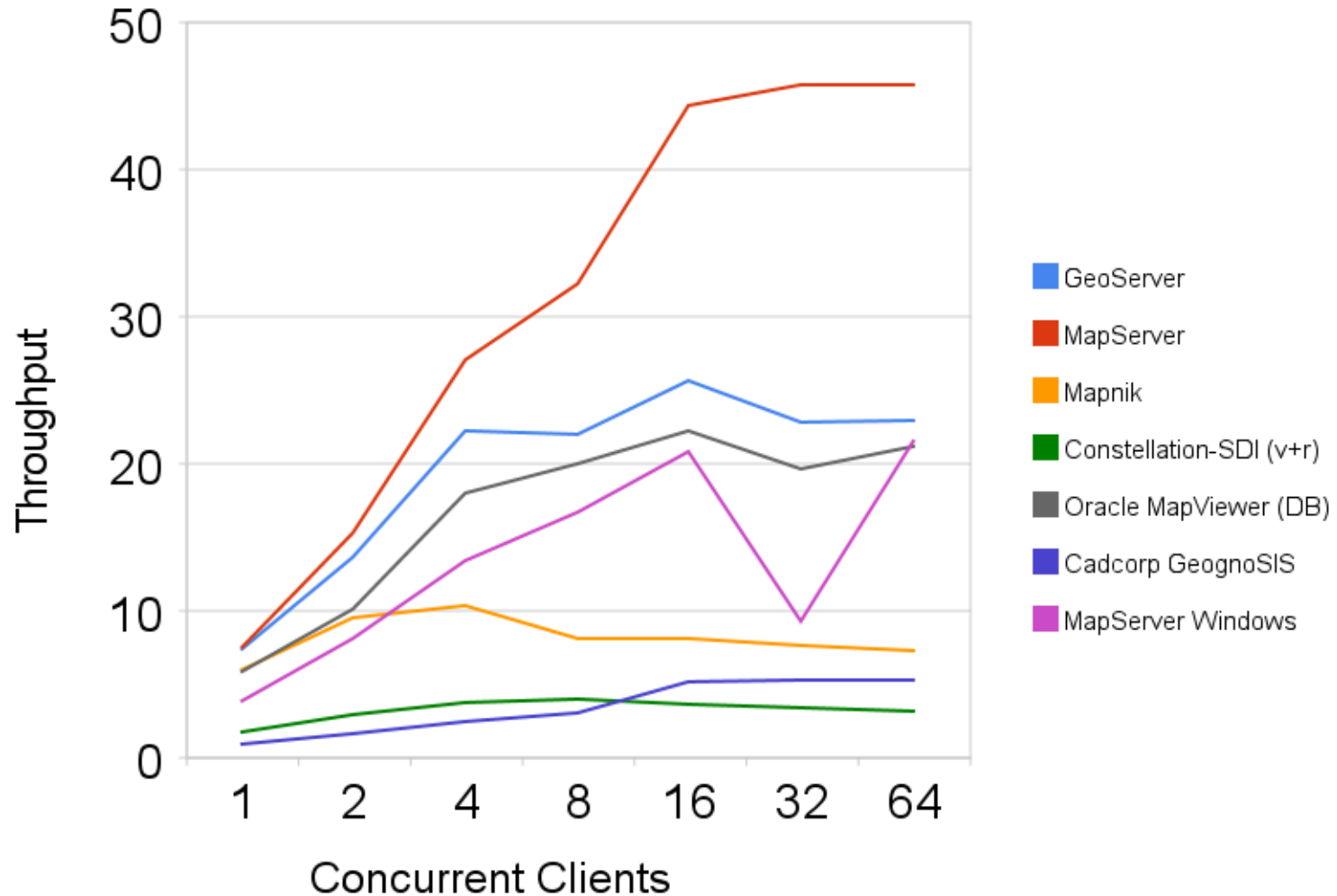






# Vector – Reprojection (EPSG:3857)

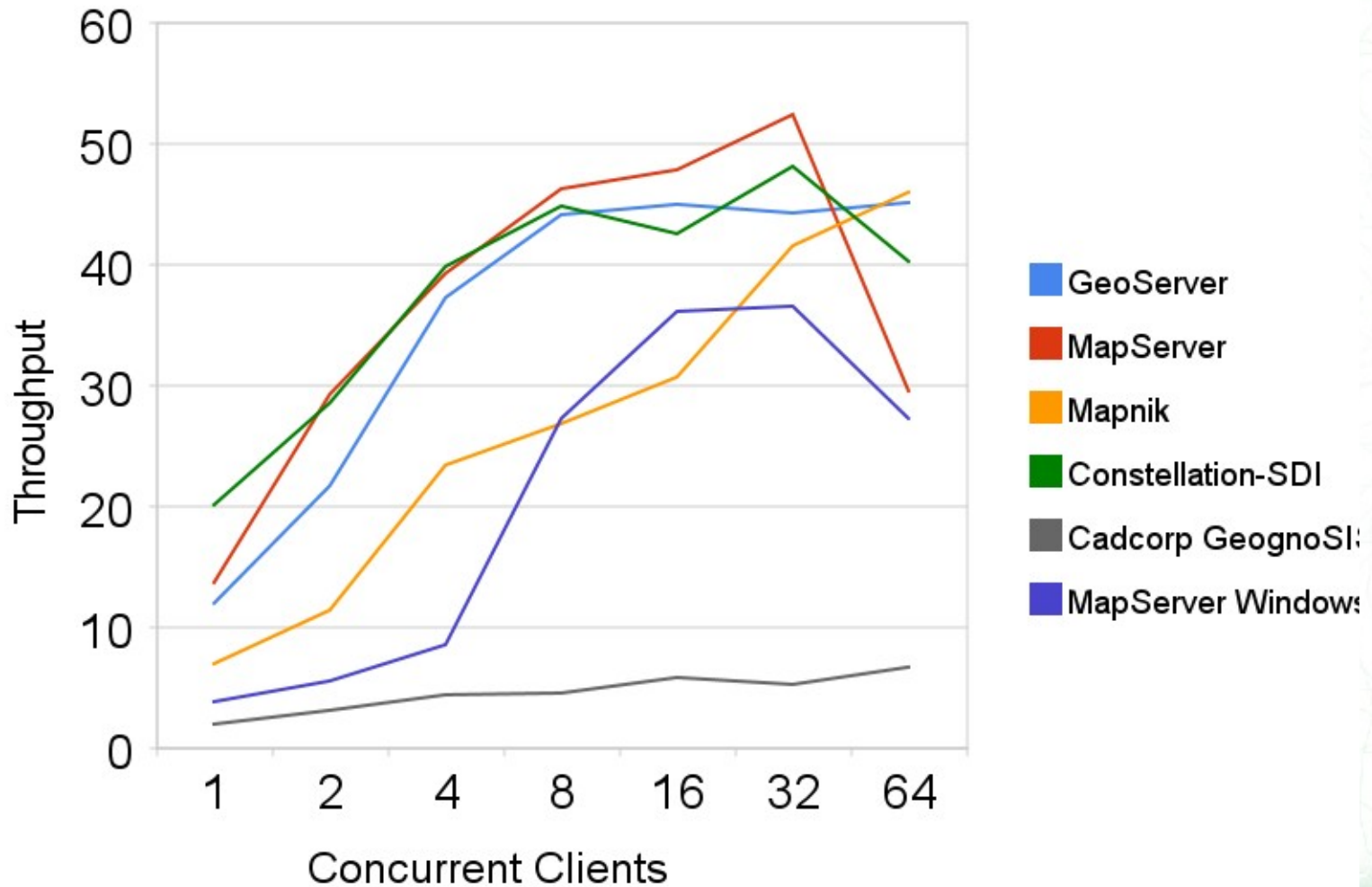
## Shapefile (reprojection)





# Raster – No Reprojection (EPSG:25831)

## GeoTIFF

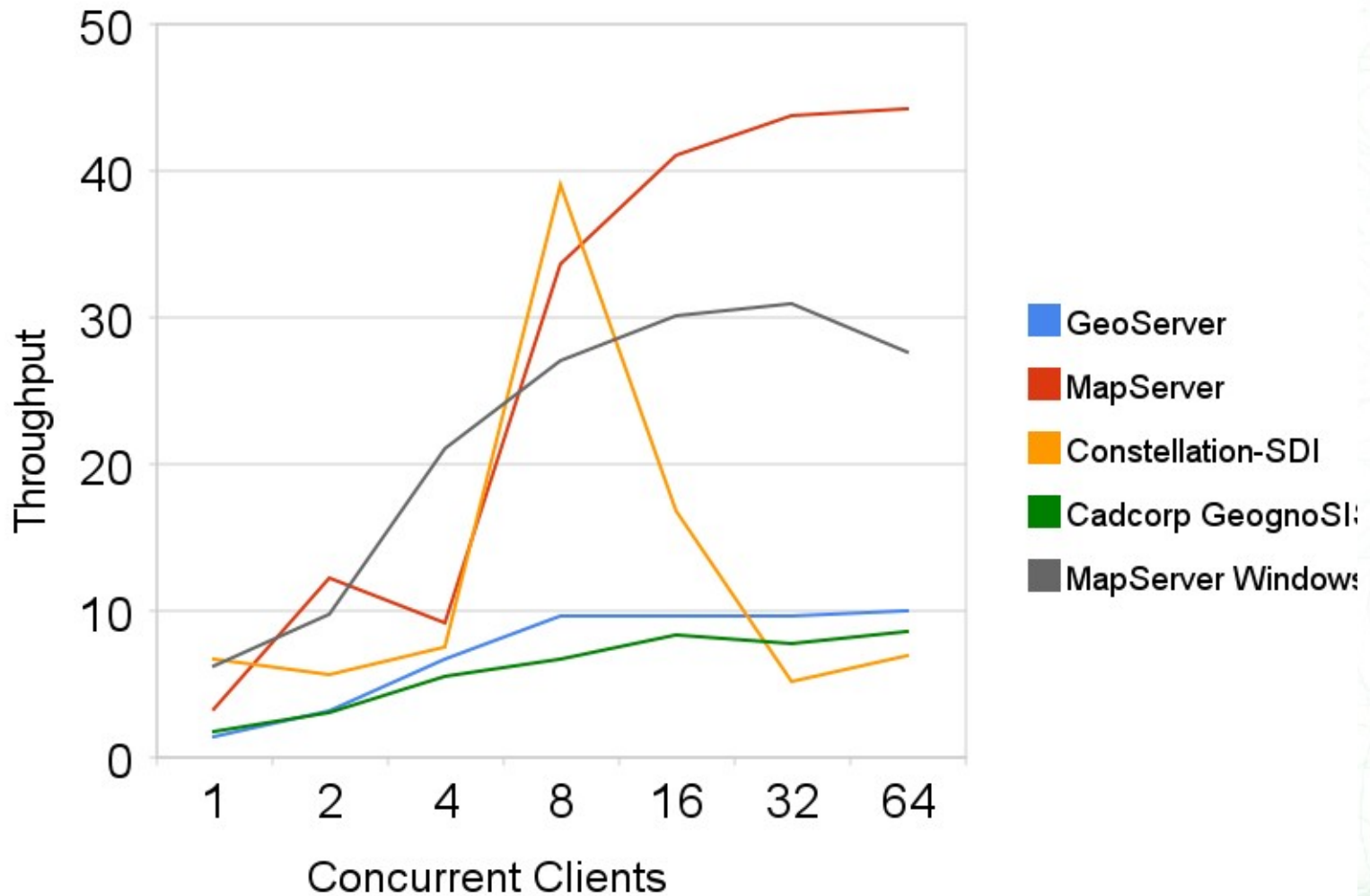






# Raster – Reprojection (EPSG:3857)

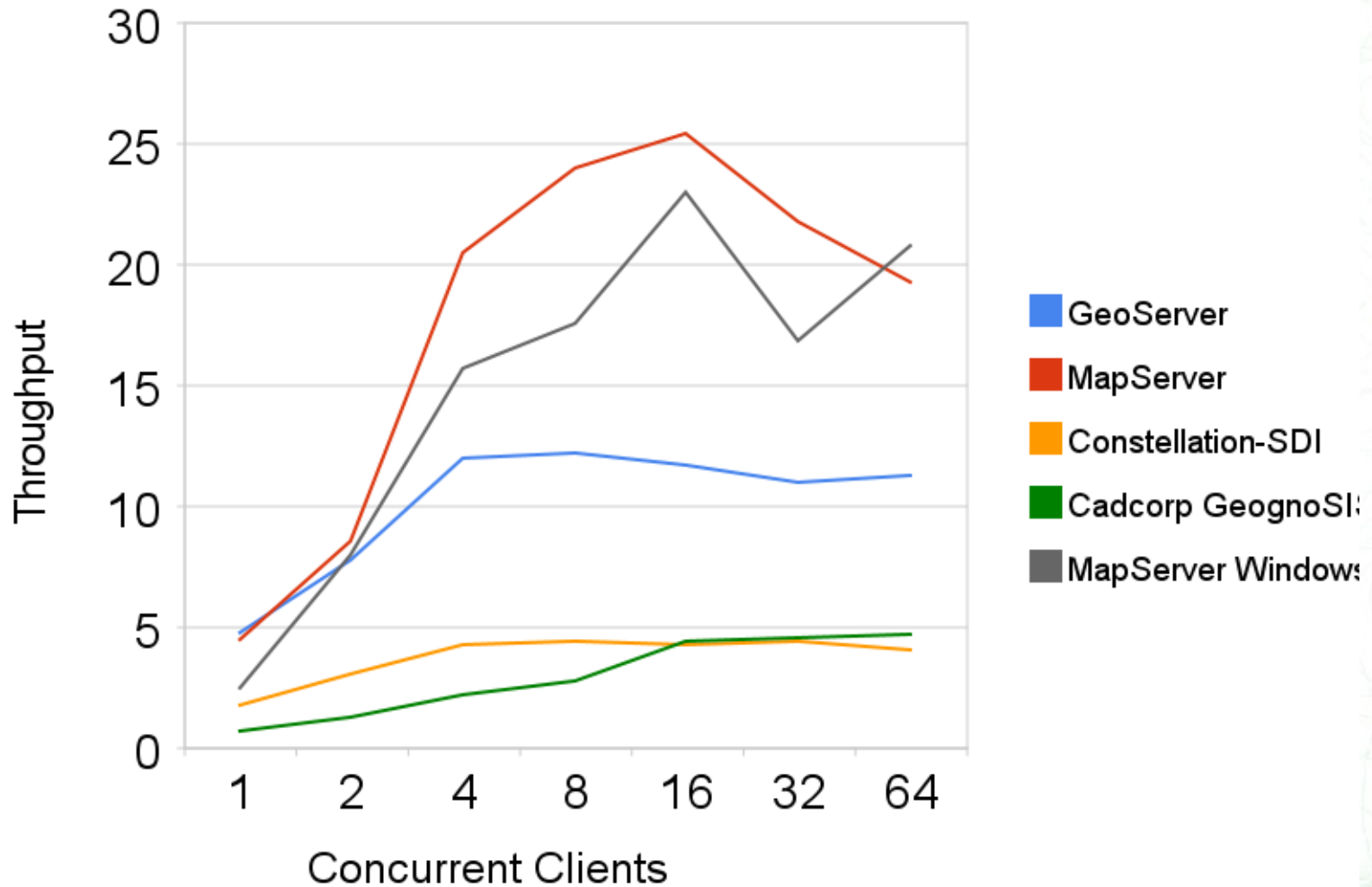
## GeoTIFF (reprojection)





# Vector/Raster – No Reproj (EPSG:25831)

## Shapefile (reprojection) + GeoTIFF

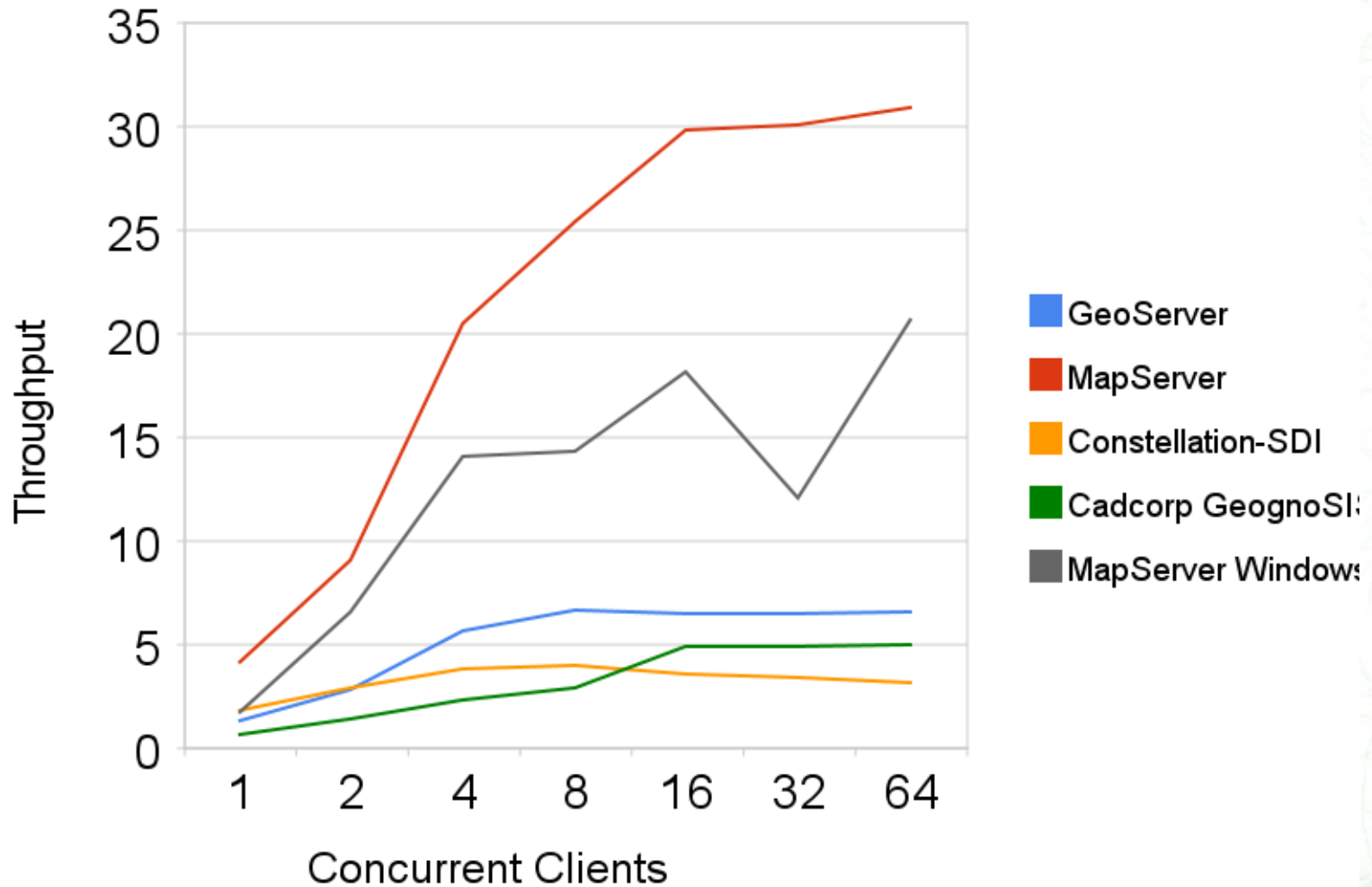






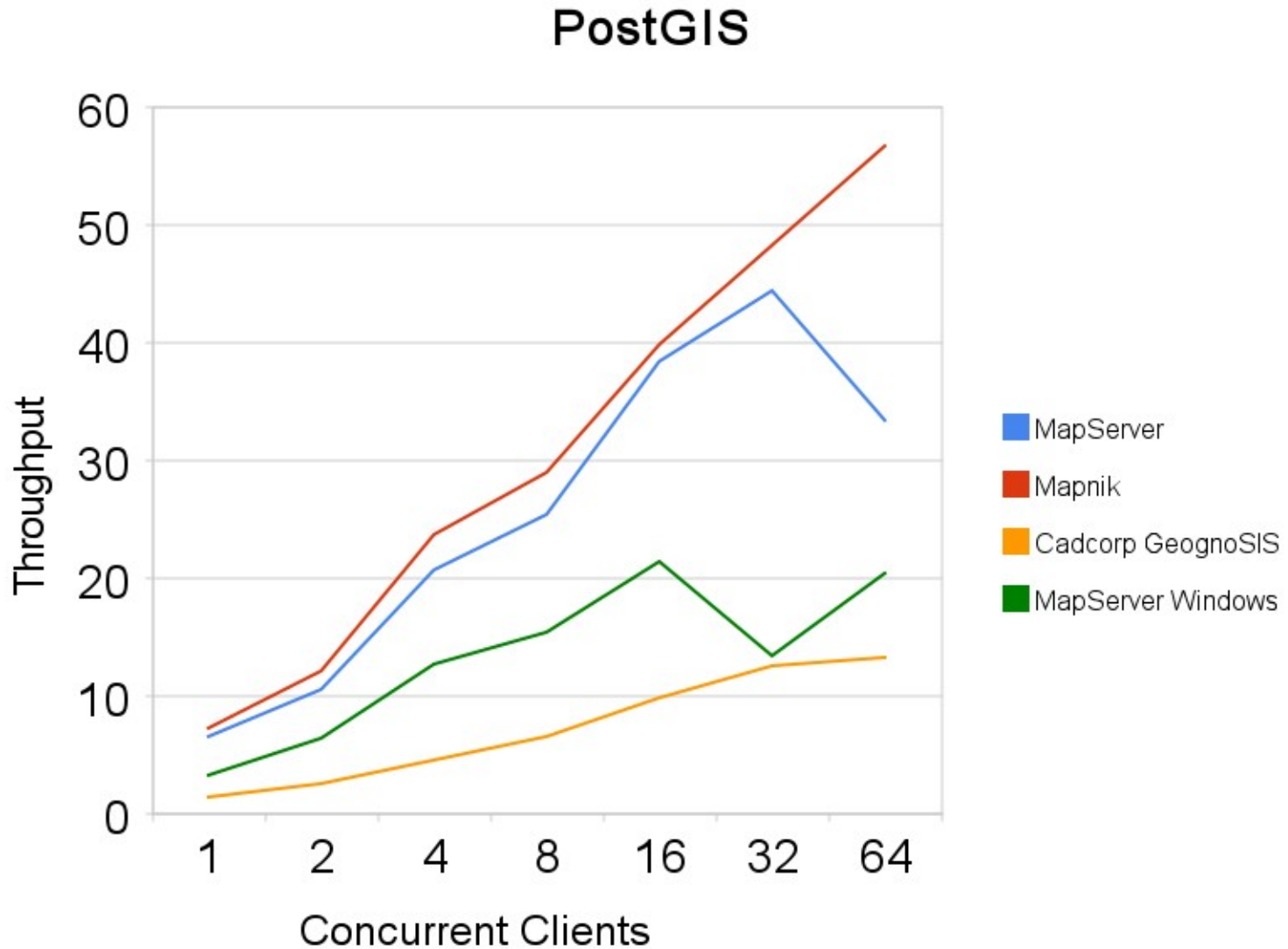
# Vector/Raster – Reproj (EPSG:3857)

## Shapefile + GeoTIFF (reprojection)





# Vector – PostGIS – No Reproj (EPSG:4326)

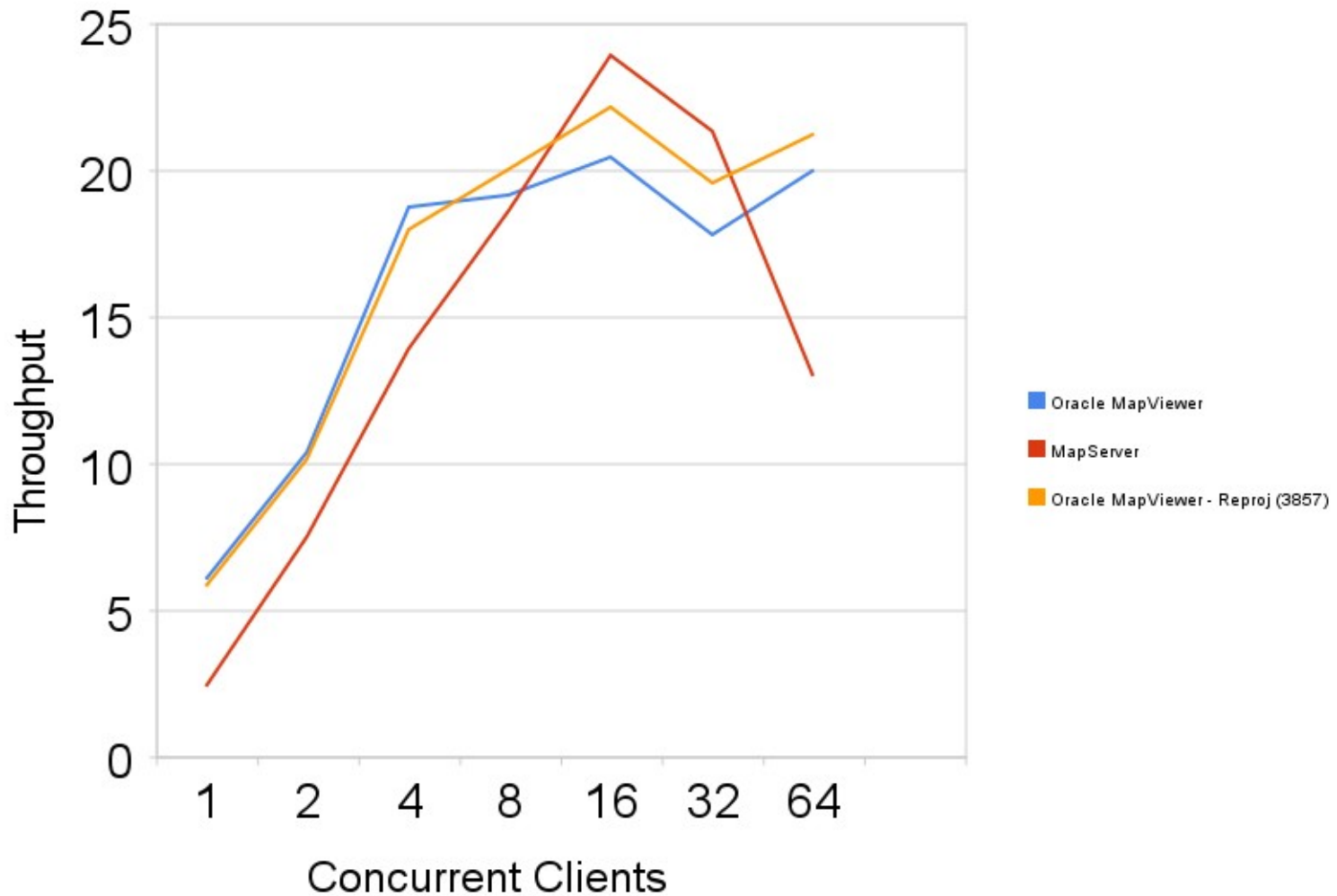






# Vector – Oracle – No Reproj (EPSG:4326)

## Oracle 11gR2 Spatial





# Hopeful Changes for Next Shootout

- Servers available early in the process (now)
- Data available VERY early in the process
- Initial test runs performed early in the process
- More participation/review throughout entire process
- Design of testing parameters agreed upon early on





## 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\\_2010](http://wiki.osgeo.org/wiki/Benchmarking_2010)
- Mailing list:  
<http://lists.osgeo.org/mailman/listinfo/benchmarking>



# Post Presentation Slides





# FLOSS4G benchmark: Navigation Movie

- The following screen demo was captured on 8th September when no other testing was taking place.
- The test client is built using OpenLayers widgets
- All maps are synchronized to the top left master.
- There is no delay in requesting maps so it is a true performance indicator based on when maps are returned
- Two servers are running on the Windows box
  - MapServer Windows
  - ERDAS APOLLO
- The two other servers are running on Red Hat Enterprise Linux
  - MapServer RHEL
  - GeoServer
- This movie also demonstrates the efforts that were done by each team to reach rendering quality, and styling compliance with the benchmark specifications