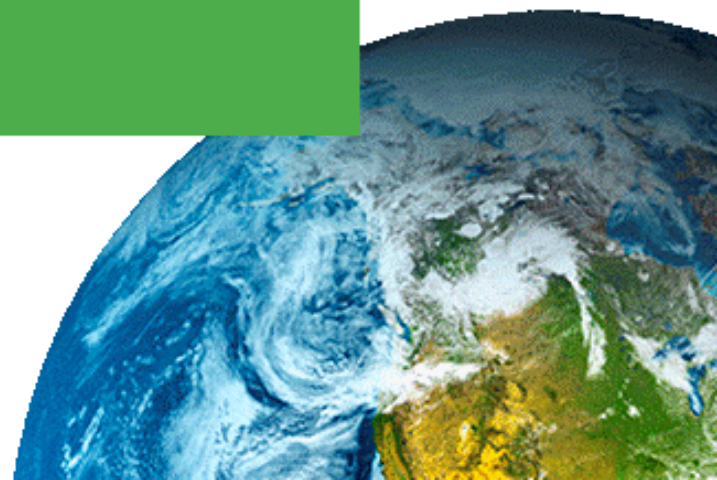


GeoServer in production

We do it, here it is how!

Ing. Andrea Aime
Ing. Simone Giannecchini
GeoSolutions



GeoSolutions

- **Founded in Italy in late 2006**
- **Expertise**
 - Image Processing, GeoSpatial Data Fusion
 - Java, Java Enterprise, C++, Python
 - JPEG2000, JPIP, Advanced 2D visualization
- **Supporting/Developing FOSS4G projects**
 - GeoServer, MapStore
 - GeoNetwork, GeoNode, Ckan
- **Clients**
 - Public Agencies
 - Private Companies
- **<http://www.geo-solutions.it>**



Contents

- Intro
- Preparing raster data
- Preparing vector data
- Optimizing styling
- Output tuning
- Tiling and caching
- Resource control
- Deploy considerations
- In Production care





GeoSolutions
your one-stop-shop for geospatial open source software



Intro



FOSS4G 2017, Boston
August 14th-19th 2017

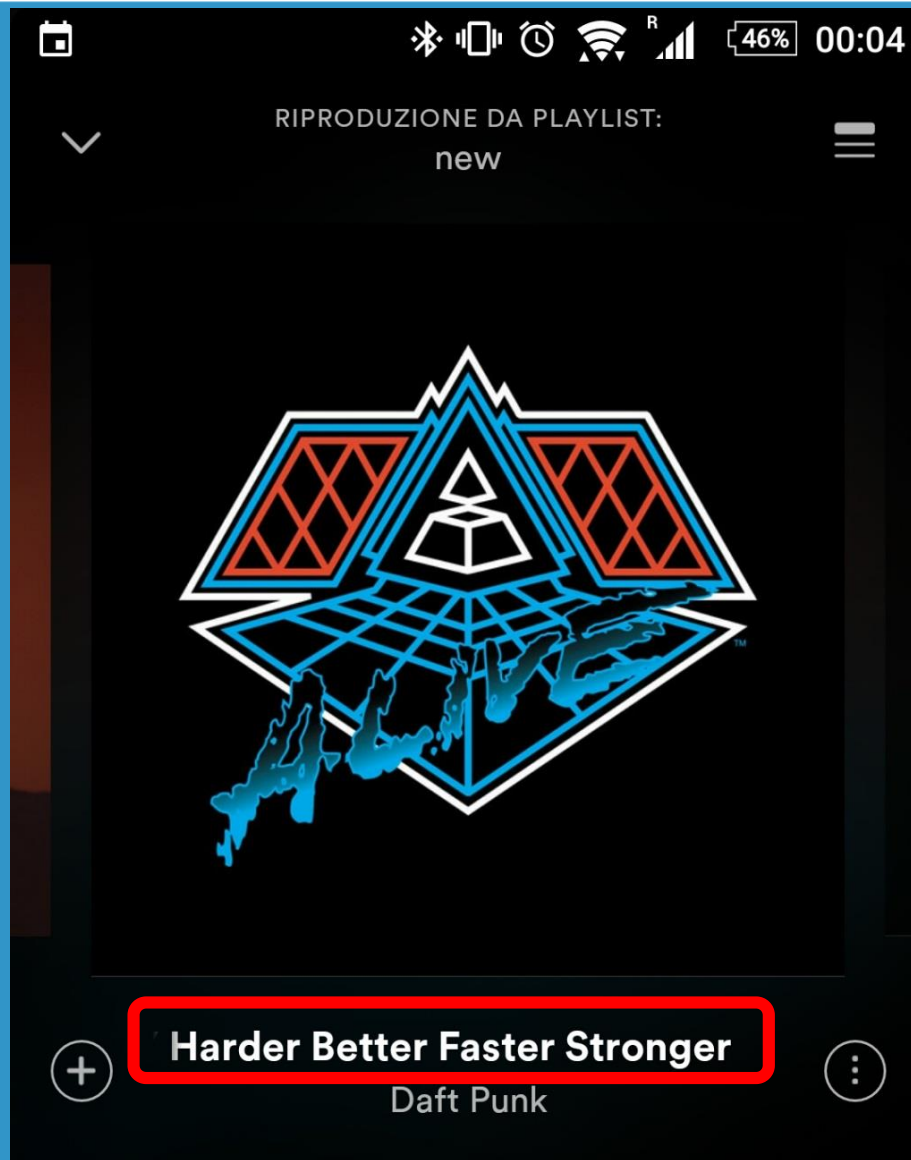


Basic Info

- **GeoServer release model**
 - Time boxed
 - Development
 - Stable
 - Maintenance
- **Don't be scared by nightly releases**
- **Basic Optimizations**
 - Study your data
 - Study your users
 - Study the deployment environment
- **Using your brain is the first optimization**



Objectives



<https://www.youtube.com/watch?v=GDpmVUEjagg>



Data preparation objectives



- Objectives
 - Fast extraction of a subset of the data
 - Fast extraction at the desired scale denominator/zoom level
- Check-list
 - Avoid having to open a large number of files per request
 - Avoid parsing of complex structures and slow compressions
- Get to know your bottlenecks
 - CPU vs Disk Access Time vs Memory





GeoSolutions
your one-stop-shop for geospatial open source software



Preparing raster inputs



FOSS4G 2017, Boston
August 14th-19th 2017



Problematic input formats



- PNG/JPEG direct serving
 - Bad formats (*especially in Java*)
 - No tiling (or rarely supported)
 - Chew a lot of memory and CPU for decompression
 - Mitigate with external overviews
- Any input ASCII format (GML grid, ASCII grid)
- ECW, fast, compresses well, but...
 - Did you know you have to buy a license to use it on server side software?





JPEG 2000 on the rise (but...)



- Becoming more and more popular with satellite imagery
- Extensible and rich, not (always) fast, can be difficult to tune for performance (*might require specific encoding options*)
- For now, fast serving at scale requires a proprietary library (Kakadu)
- But keep an eye on OpenJPEG, effort underway to make it faster/use less memory:

<http://www.openjpeg.org/>



GeoTIFF for the win



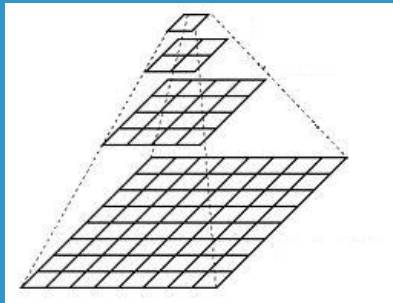
- To remember: GeoTiff is a swiss knife
 - *But you don't want to cut a tree with it!*
 - Tremendously flexible, good for for most (*not all*) use cases
 - BigTiff pushes the GeoTiff limits farther
- Use GeoTiff when
 - Overviews and Tiling stay within 4GB
 - No additional dimensions
- Consider BigTiff for very large file (> 4 GB)
 - Support for tiling
 - Support for Overviews
 - Can be inefficient with very large files + small tiling



Possible structures

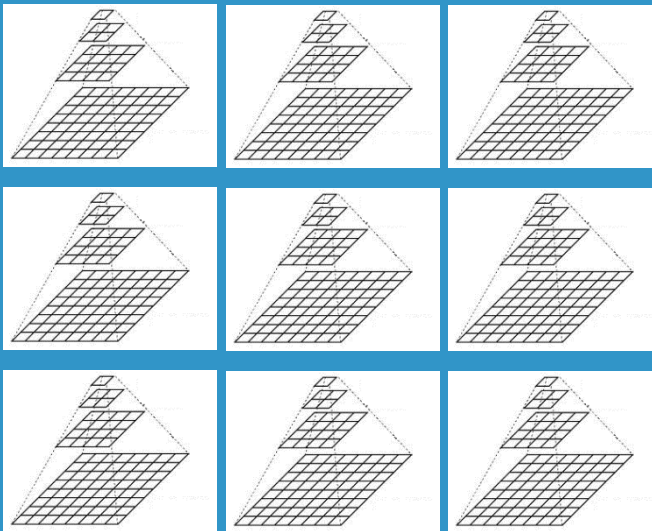


1



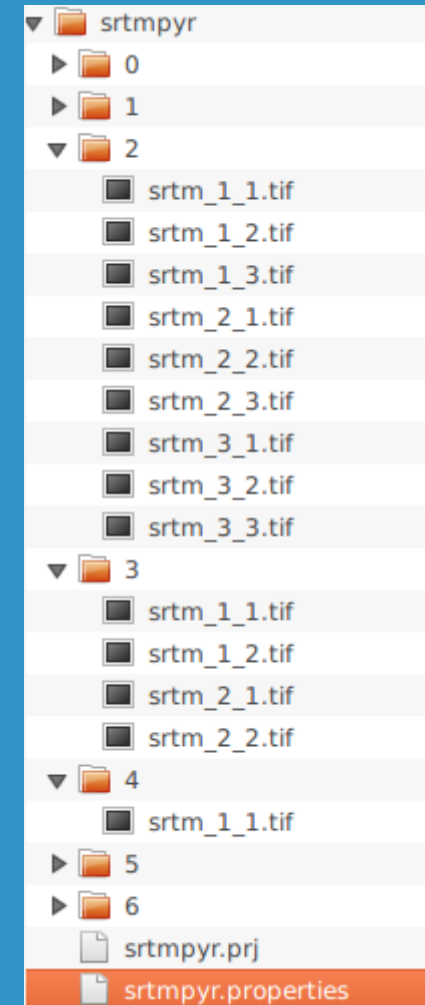
Single GeoTiff
with internal tiling
and overviews
(GeoTiff < 2GB,
BigTiff < 20-50GB)

2



Mosaic of GeoTiff, each one
with internal tiling and overviews
(< 500GB, not too many files)

3



Pyramid

Choosing formats and layouts



- For single granules (< 20Gb) GeoTiff is generally a good fit
- Use ImageMosaic when:
 - A single file gets too big (inefficient seeks, too much metadata to read, etc..)
 - Multiple Dimensions (time, elevation, others..)
 - Avoid mosaics made of many very small files
 - Single granules can be large
 - Use Tiling + Overviews + Compression on granules
- Use ImagePyramid when:
 - Tremendously large dataset
 - Too many files / too large files
 - Need to serve at all scales
 - Especially low resolution



Raster data preparation



- Re-organize (merge files, create pyramid, reproject)
- Compress (eventually)
- Retile, add overviews
- Get all the details in our training material:

http://geoserver.geo-solutions.it/edu/en/raster_data/index.html

The screenshot shows a slide from a GeoServer Training course. At the top, there is the GeoSolutions logo and tagline: "GeoSolutions your one-stop-shop for geospatial open source software". Below this, a blue box contains the text "GeoServer Training". The main title of the slide is "Advanced Raster Data Management". Underneath the title, there is a bulleted list of topics:

- [Introduction To Processing With GDAL Utilities](#)
- [Advanced Mosaics and Pyramids Configuration](#)
- [Using ImageMosaic with Footprint](#)
- [Advanced Processing With GDAL Utilities](#)





GeoSolutions
your one-stop-shop for geospatial open source software



Preparing vector inputs



FOSS4G 2017, Boston
August 14th-19th 2017



Choosing a format



- Slow formats, text based, not indexed
 - WFS
 - GML
 - DXF
 - CSV
 - GeoJSON
- Good formats, local and indexable
 - Shapefile
 - GeoPackage
 - Spatial databases: PostGIS, Oracle Spatial, DB2, SQL server, MySQL
 - NoSQL: SOLR, MongoDB, ...





- Choose PostGIS if you can, it has the best query planner for spatial and plans every query based on the query parameter (GIS makes for wildly different optimal plans depending on the bbox you queried)
- Rich support for complex native filters
- Use connection pooling
- Validate connections (*with proper pooling*)
- Table Clustering
- Spatial and Alphanumeric Indexing
- Spatial and Alphanumeric Indexing
- Spatial and Alphanumeric Indexing
- ...
- *Did we mention indexes?*



Connection pooling tricks



- Connection pool size should be proportional to the number of concurrent requests you want to serve (obvious no?)
- Activate connection validation
- Mind networking tools that might cut connections sitting idle (yes, your server is not always busy), they might cut the connection in “bad” ways (10 minutes timeout before the pool realizes the TCP connection attempt gives up)
- Read more
 - [Advanced Database Connection Pooling Configuration](#)
 - [DBMS Connections Params Explained](#)



Shapefile preparation



- Remove .qix file if present (GeoServer will recreate it, sometimes better optimized)
- If there are large DBF attributes that are not in use, get rid of them using ogr2ogr, e.g.:

```
ogr2ogr -select FULLNAME,MTFCC arealm.shp
```

```
tl_2010_o8013_arealm.shp
```

- *If on Linux*, enable memory mapping, faster, more scalable (but will kill Windows):

- Create spatial index if missing/outdated
- Use memory mapped buffers
- Cache and reuse memory maps



Shapefile filtering



- Stuck with shapefiles and have scale dependent rules like the following?
 - Show highways first
 - Show all streets when zoomed in
- Use ogr2ogr to build two shapefiles, one with just the highways, one with everything, and build two layers, e.g.:

```
ogr2ogr -sql "SELECT * FROM  
t1_2010_08013_roads WHERE MTFCC in  
( 'S1100', 'S1200' )" primaryRoads.shp  
t1_2010_08013_roads.shp
```

- Or hire us to develop non-spatial indexing for shapefile!





GeoSolutions
your one-stop-shop for geospatial open source software



Optimize styling



FOSS4G 2017, Boston
August 14th-19th 2017



Use scale dependencies



- Never show too much data
 - the map should be readable, not a graphic blob. Rule of thumb: 1000 features max in the display
 - Show details as you zoom in



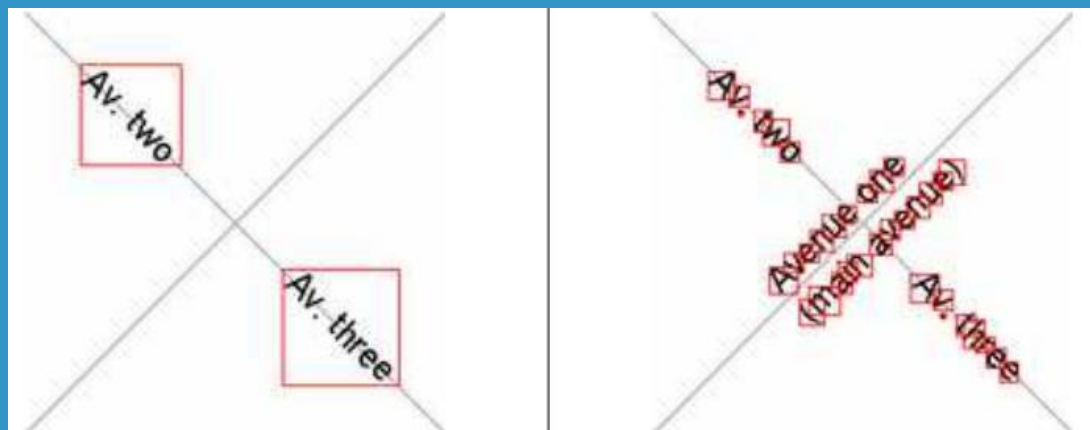
- Eagerly add `MinScaleDenominator` to your rules
- Add more expensive rendering when there are less features
- Key to get both a good looking and fast map



Labeling



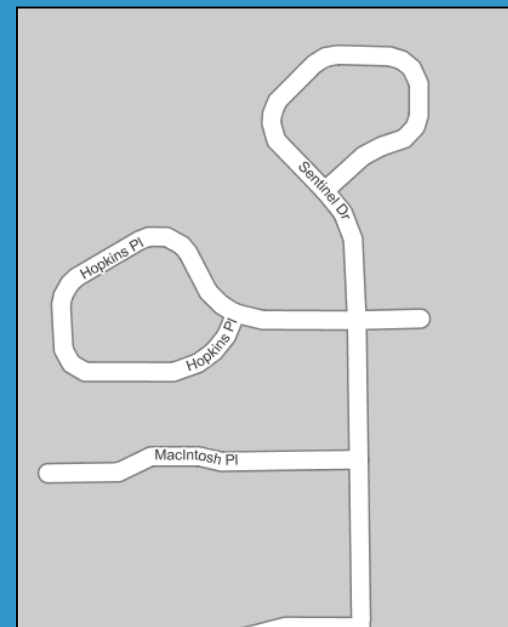
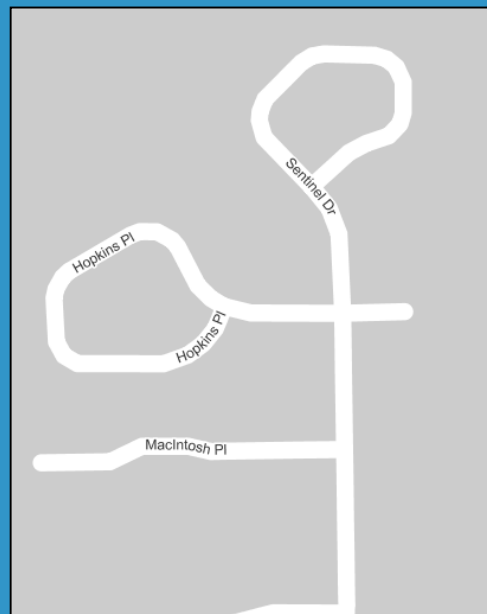
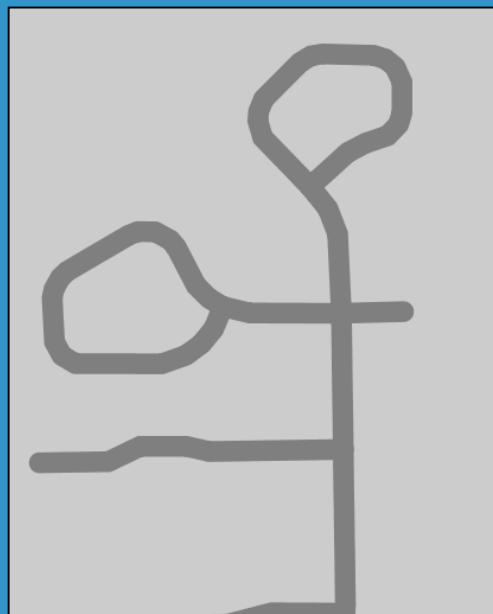
- Labeling conflict resolution is expensive, limit to the most inner zooms
- Careful with maxDisplacement, makes for various label location attempts
- GeoServer 2.9 onwards has per char space allocation, much better looking labelling, but more expensive too, disable if in dire need via sysvar
–Dorg.geotools.disableLetterLevelCache=true



FeatureTypeStyle

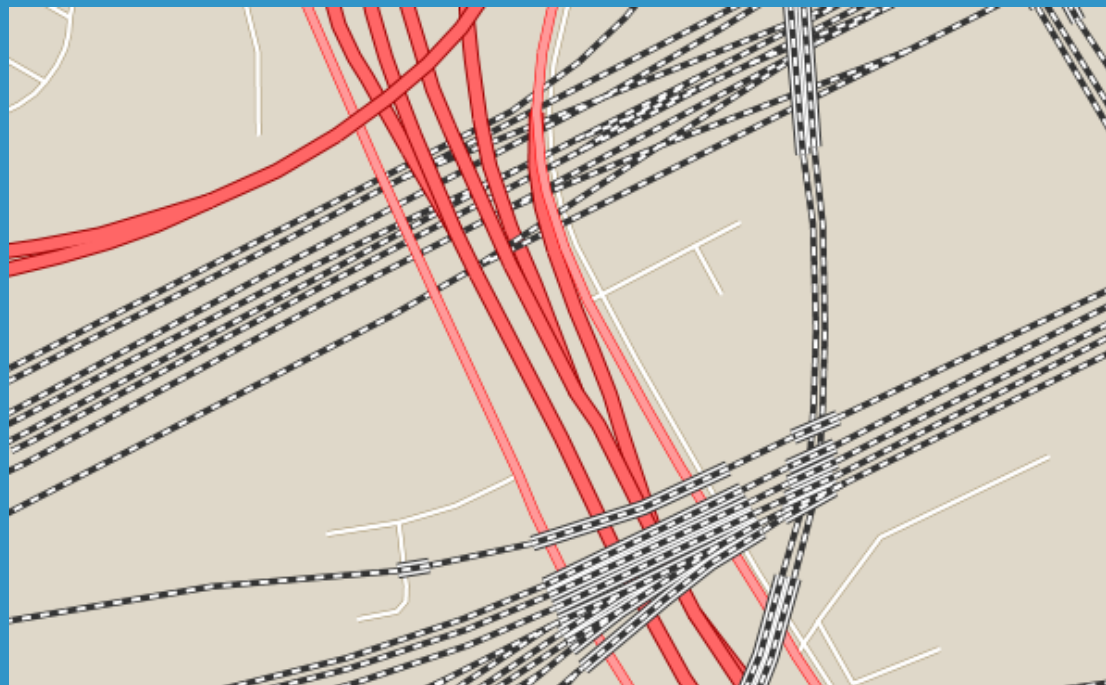


- GeoServer uses SLD FeatureTypeStyle objects as Z layers for painting
- Each one allocates its own rendering surface (which can use a lot of memory), use as few as possible





- Use DBMS as the data source
- Add indexes on the fields used for z-ordering
- If at all possible, use cross-feature type and cross-layer z-ordering on small amounts of data (we need to go back and forth painting it)





GeoSolutions
your one-stop-shop for geospatial open source software



Tiling and caching



FOSS4G 2017, Boston
August 14th-19th 2017



Tile caching with GWC

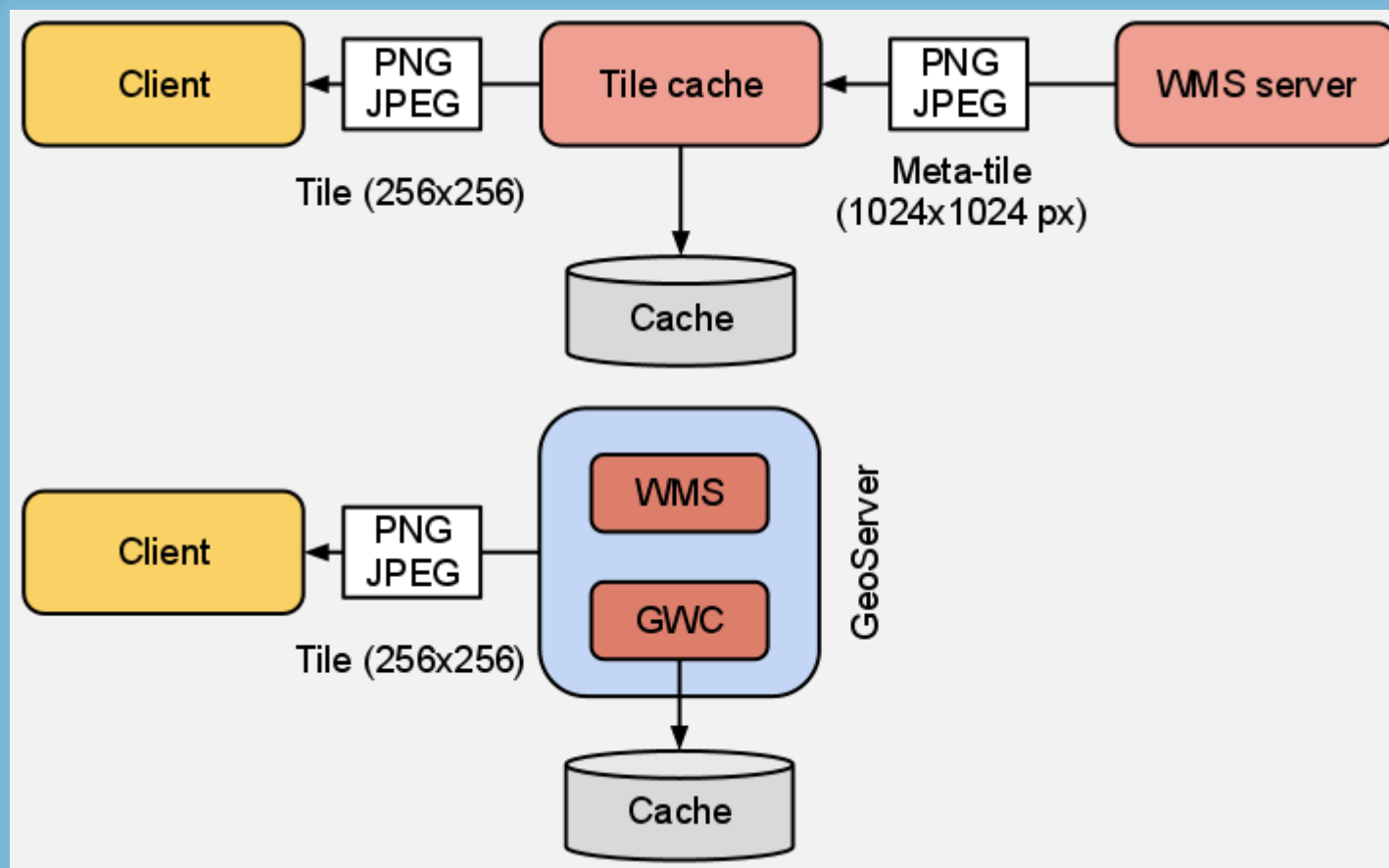


- Tile oriented maps, fixed zoom levels and fixed grid
- Useful for stable layers, backgrounds
- Protocols: WMTS, TMS, WMS-C, Google Maps/Earth, VE
- Speedup compared to dynamic WMS: 10 to 100 times, *assuming tiles are already cached* (whole layer pre-seeded)
- Suitable for:
 - Mostly static layer
 - No/few dynamic parameters (CQL filters, SLD params, SQL query params, time/elevation, format options)





- No double encoding when using meta-tiling, faster seeding



Space considerations



- Seeding Colorado, assuming 8 cores, one layer, 0.1 sec 756x756 metatile, 15KB for each tile
- Do yours: <http://tinyurl.com/3apkpss>
- Not enough disk space? Set a disk quota

Zoom level	Tile count	Size (MB)	Time to seed (hours)	Time to seed (days)
13	58,377	1	0	0
14	232,870	4	0	0
15	929,475	14	0	0
16	3,713,893	57	1	0
17	14,855,572	227	6	0
18	59,396,070	906	23	1
19	237,584,280	3,625	92	4
20	950,273,037	14,500	367	15



Client side cache



- Make client not request tiles, use their local cache instead
- HTTP headers, time to live, eTag
- Does not work with browsers in private mode

```
<expireClientsList>  
  <expirationRule minZoom="0" expiration="7200" />  
  <expirationRule minZoom="10" expiration="600" />  
</expireClientsList>
```



Headers Preview Response Cookies Timing

Host: tms.comune.+1.1t
Referer: http://datigis.comune.fi.it/MapStore/
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/24.0.1313.107 Safari/537.36

▼ **Query String Parameters** [view URL encoded](#)

SERVICE: WMS
STYLES:
FORMAT: image/png
TRANSPARENT: true
LAYERS: ambiente:areeverdi
EXCEPTIONS: application/vnd.ogc.se_xml
VERSION: 1.1.1
REQUEST: GetMap
SRS: EPSG:900913
BBOX: 1245929.9245239,5430629.9245239,1247152.9169763,5430629.9245239
WIDTH: 256
HEIGHT: 256

▼ **Response Headers** [view source](#)

Cache-Control: max-age=604800, must-revalidate
Connection: Keep-Alive
Content-Length: 8943
Content-Type: image/png
Date: Sat, 26 May 2012 09:43:51 GMT
Expires: Sat, 02 Jun 2012 09:43:51 GMT
Keep-Alive: timeout=15, max=98
geowebcache-cache-result: HIT
geowebcache-tile-index: [10, 10, 14]



Choose the right format



- Use the right formats:
 - JPEG for background data (e.g. ortos)
 - PNG8 + precomputed palette for background vector data (e.g. basemaps)
 - PNG8 full for vector overlays with transparency
 - image/vnd.jpeg-png for raster overlays with transparency
- The format impacts also the disk space needed! (as well as the generation time)
- Check this [blog post](#)



Vector tiles



- Extension to support vector tiles
- Still new, but:
 - PNG encoding is often 50% of the request time when there is little data in the tile
 - Gone with Vector tiles
 - Vector tiles allow over-zooming, meaning you can build less zoom levels (reducing the total size by a factor of 4 or 16)
 - Vector tiles are more compact
 - However, not an OGC/ISO standard





MBTiles Cache Store Support



- New community module
 - Available from 2.9.2
 - Doc [here](#)
- File Path Templates
 - {layer}/{grid}{format}{params}/{z}-{x}-{y}.sqlite
 - {layer}/{grid}{format}{params}/standardMbTiles.sqlite
- Replace Operation
 - **Seed offline then replace online!**
- Coupled with in memory caching delivers high performance!



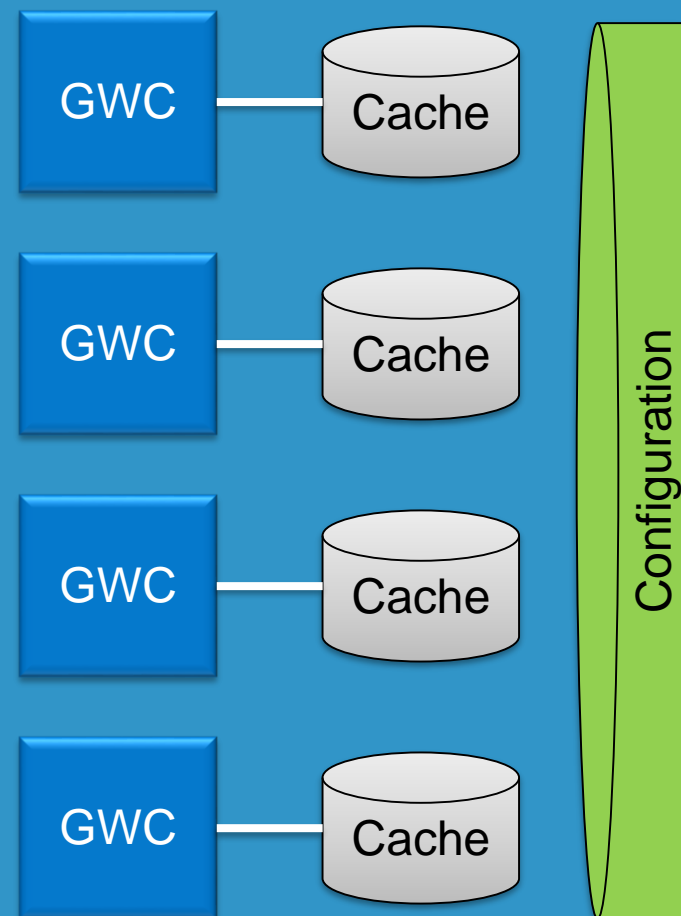
FOSS4G 2017, Boston
August 14th-19th 2017



Independent Caches Option



- Each node in the cluster is given its own cache on local disk
- Trading disk occupation for speed
- Good fit for clustered environments
 - Network file systems may suffer under write load
- Shared Configuration
- Independent operations
- Especially valuable for dynamic, non fully seeded caches





GeoSolutions
your one-stop-shop for geospatial open source software



Resource control

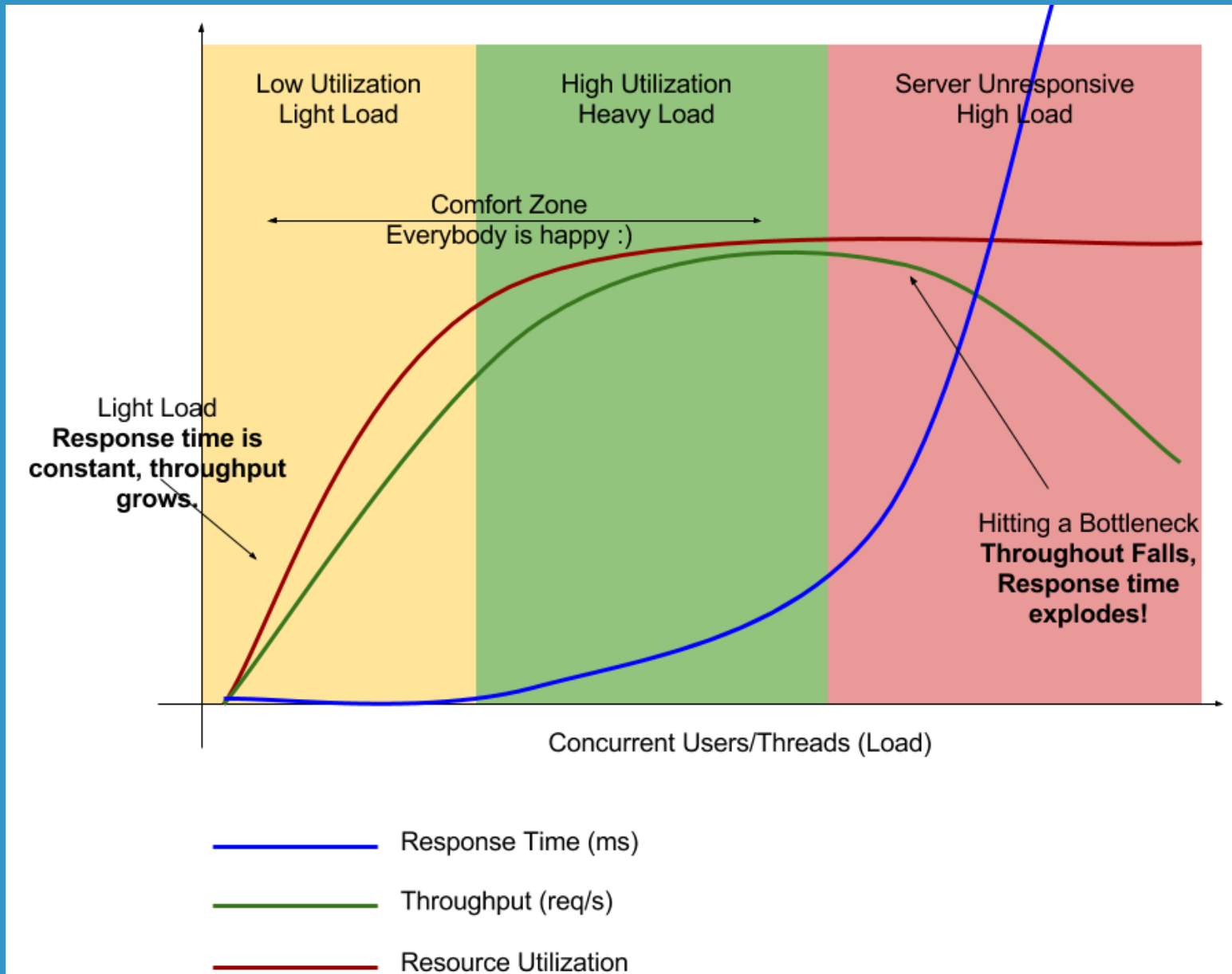


FOSS4G 2017, Boston
August 14th-19th 2017





What happens on your server



Set the Resource Limits



- Limit the amount of resources dedicated to an individual request
- Improve fairness between requests, by preventing individual requests from hijacking the server and/or running for a very long time
- EXTREMELY IMPORTANT in production environment
- WHEN TO TWEAK THEM?
 - Frequent OOM Errors despite plenty of RAM
 - Requests that keep running for a *long* time (e.g. CPU usage peaks even if no requests are being sent)
 - DB Connection being killed by the DBMS while in usage (ok, you might also need to talk to the DBA..)





Resource limits per service



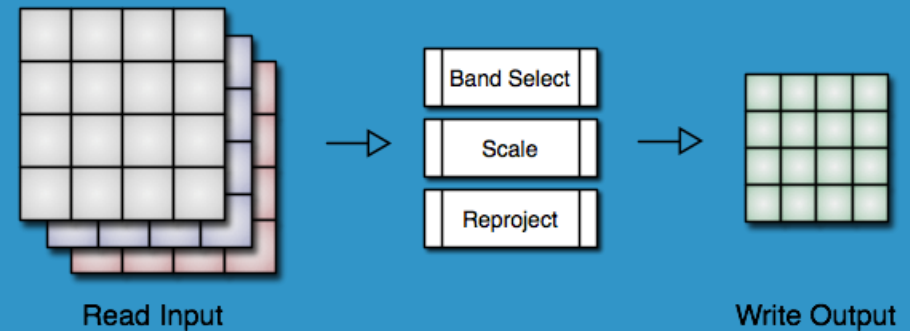
Resource consumption limits

Max rendering memory (KB)
65536

Max rendering time (s)
120

Max rendering errors (count)
1000

WMS



Features

Maximum number of features
1000000

Return bounding box with every feature

WFS

Resource Consumption Limits

Maximum amount of data read (KB, 0 for no limit)
100000

Maximum amount of data generated (KB, 0 for no limit)
100000

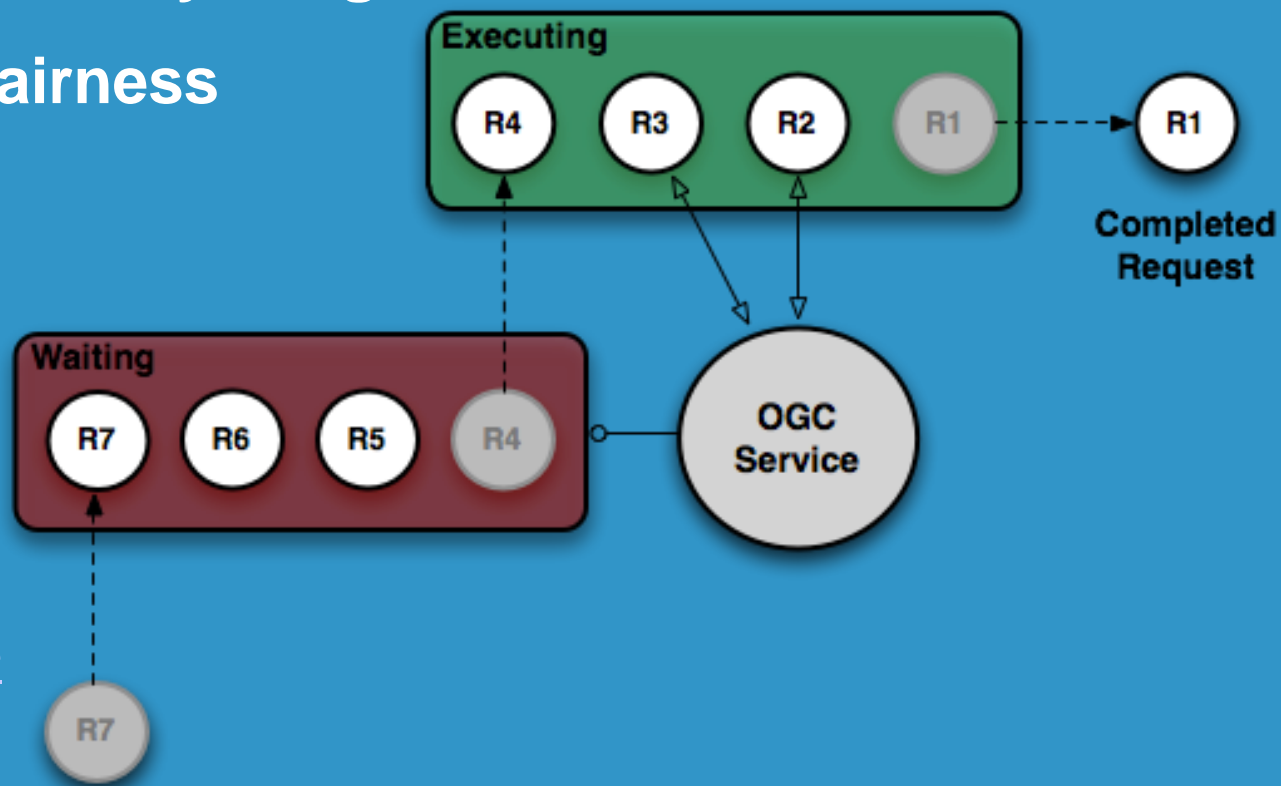
WCS



Control-flow



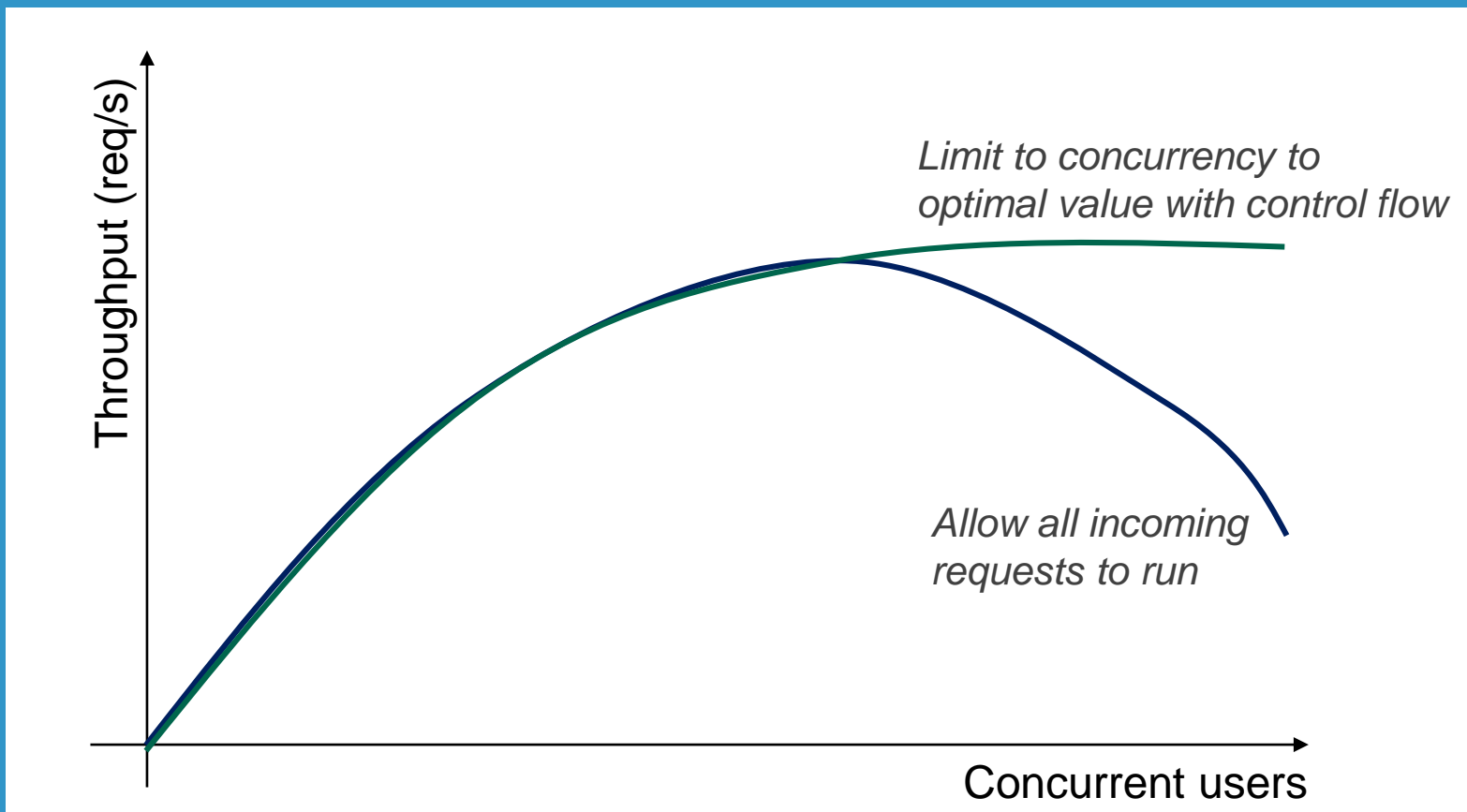
- Control how many requests are executed in parallel, queue others:
 - Increase throughput
 - Control memory usage
 - Enforce fairness



- More info [here](#)



Control-flow



`$GEOSERVER_DATA_DIR/controlflow.properties`

```
# don't allow more than 16 GetMap requests in parallel  
ows.wms.getmap=16
```





JVM and deploy configuration



Go back and optimize the rest first



- There is no “GO FAST!” option in the Java Virtual Machine
- The options discussed here are not going to help if you did not prepare the data and the styles
- They are finishing touches that can get performance up once the major data bottlenecks have been dealt with
- Check “Running in production” instructions [here](#)



JVM settings



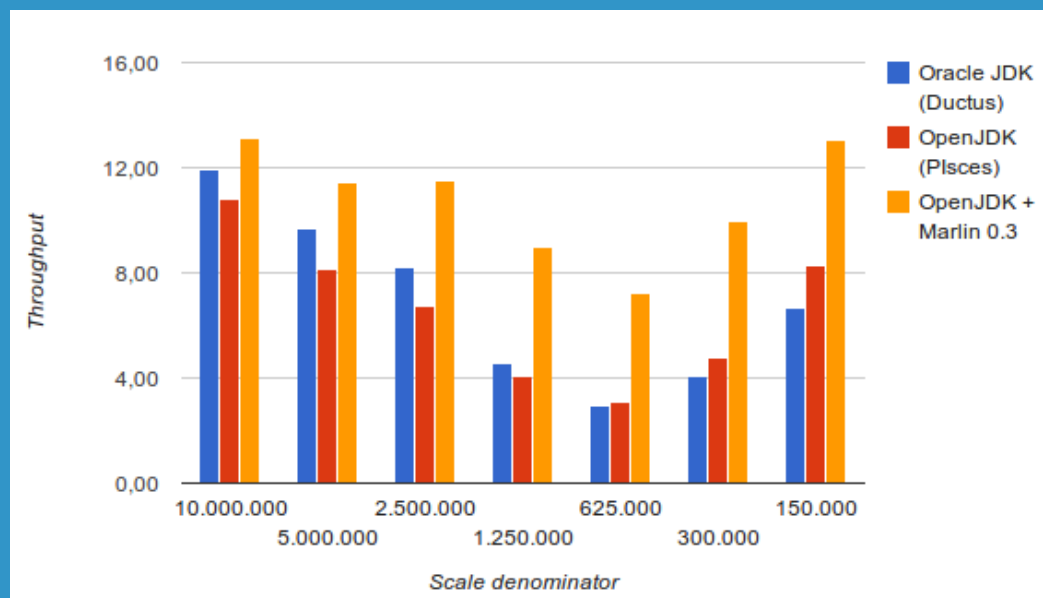
- **`--server`**: enables the server JIT compiler
- **`--Xms2048m -Xmx2048m`**: sets the JVM use two gigabytes of memory
- **`--XX:+UseParallelOldGC -XX:+UseParallelGC`**: enables multi-threaded garbage collections, useful if you have more than two cores (G1 provides inconsistent results, needs more testing)
- **`--XX:NewRatio=2`**: informs the JVM there will be a high number of short lived objects
- **`--XX:+AggressiveOpt`**: enable experimental optimizations that will be defaults in future versions of the JVM



Marlin renderer



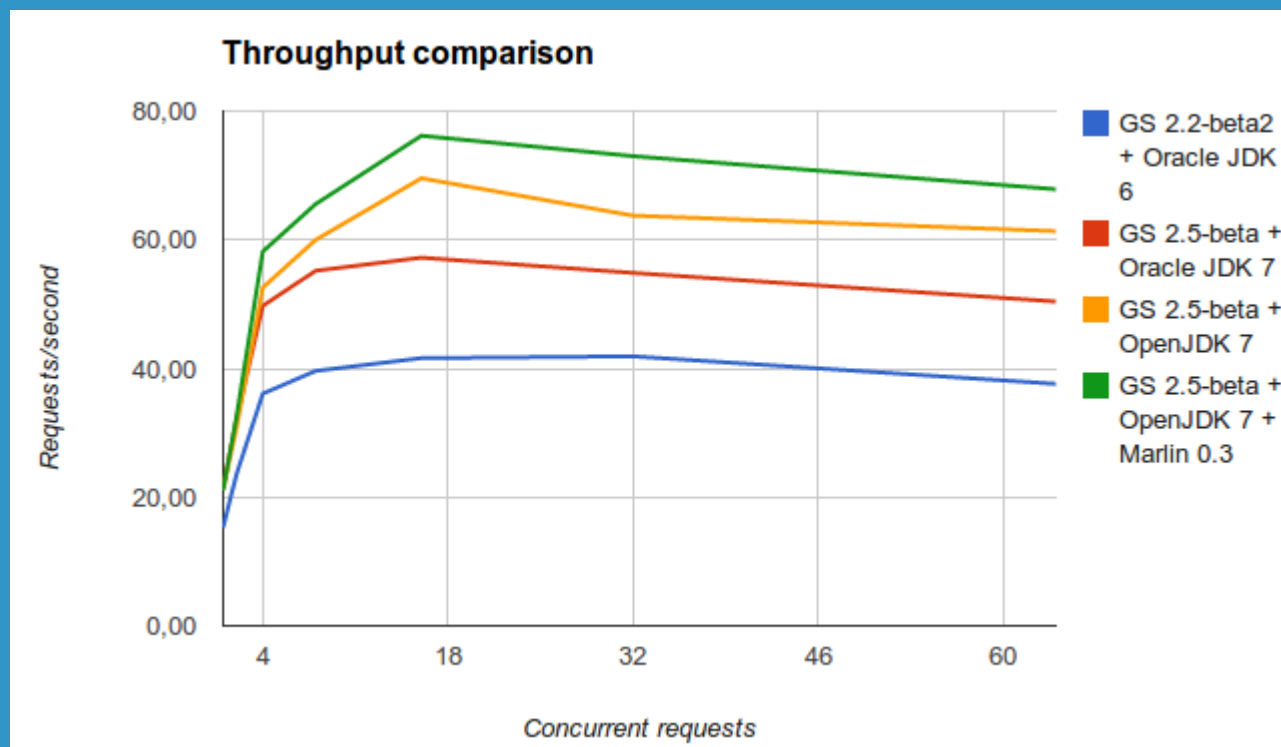
- The OpenJDK Java2D renderer scales up, but it's not super-fast when the load is small (1 request at a time)
- The Oracle JDK Java2D renderer is fast for the single request, but does not scale up
- Marlin-renderer to the rescue:
<https://github.com/bourgesl/marlin-renderer>
- *It is already the official renderer for OpenJDK 9 (beta)*
- *But for now GeoServer won't run on JDK 9!*



Upgrade!



- Performance tends to go up version by version
- Please do use a recent GeoServer version
- FOSS4G 2010 vector benchmark with different versions of GeoServer, throughput keeps on improving





JAI Settings

Administer settings related to Java Advanced Imaging.

Memory Capacity (0-1)

Memory Threshold (0-1)

Tile Threads

Tile Threads Priority

 Tile Recycling JPEG Native Acceleration PNG Native Acceleration Mosaic Native Acceleration

- Install the TurboJPEG extension
- Enable JAI Mosaicking native acceleration
- Give JAI enough memory
- Don't raise JAI memory Threshold too high
- *Rule of thumb: use 2 X #Core Tile Threads (check next slide)*
- Play with tile Recycling against your workflows (might help, might not)



We are in prod, now what?



When in production



- When the going gets tough, the tough get going!
 - Performance suboptimal
 - OOM
 - Occasional Deadlocks and Stalls
- Hang tight before reading next line...
- That is normal!
 - Don't have any of these problems that means nobody uses your services
 - Reaching PROD does not mean the work has ended!*

* hello beloved client, did you read that?



When in production



- Ok, we are in the same boat
 - Thanks, but what can I do?
- Here some key concepts
 - **Logging**
 - **Monitoring**
 - **Metering**
- You want to be able to know what happens before it actually happens*!
 - or better before someone call you on the phone screaming and shouting!





- **Logging**
- When you are sick, a good doctor should ask you how you feel, right?
- We should do the same with GeoServer
- Logs are hard to read
 - Logging levels are your friend
 - Look for known errors first
- Logs of a network exposed service are usually full of errors and exceptions
 - Unless nobody uses that service 😊



When in production



- **Monitoring**
- When you are in PROD you have to understand and monitor every bit involved
 - DBMS, Disks
 - CPU, Memory , Network
 - Other Software
- **Proactivity**
 - **Alerting** → low RAM, high cpu, low disk space
 - **Actions** → service dead/stuck then restart





When in production



GeoSolutions
your one-stop-shop for geospatial open source software

Monitoring

New Relic APM BROWSER SYNTHETICS MOBILE SERVERS PLUGINS INSIGHTS Alerts New Tools Help GeoSolutions_IT

Filter applications Off Show labels Add more

Name	End user	Page views	App server	Throughput	Error %
ComuneGE - GeoServer Cluster (Slaves)			78.4 ms	299 rpm	0%
Neftex Prod - GS					
Preview GRID UN					
Windows7 - Build					
OWS12 - GeoSer					
ComuneFI - GW					
Demo - GeoServ					
LHTAC - MapSto					
VIBI - GeoServ					
NRL Crop Inform					
NPA-CGG GeoSe					
Neftex Prod - GS					
Hart Energy Geo					
Geo3 - MapSto					

Web transactions time 235 ms APP SERVER 0 s BROWSER

Apdex score 0.92 [0.5] NS [7.0]* APP SERVER BROWSER

Throughput 167 rpm AVERAGE

Transactions	App server time	Error rate
/dispatcher	236 ms	6%
/Set Character Encoding	19.4 ms	4%

Member of policy Shut Up!

Recent events: No Events In The Last 3 Days.



FOSS4G 2017, Boston
August 14th-19th 2017





- **Monitoring**
- Troubleshooting GeoServer





- **Metering**
- **Measuring KPI is crucial**
 - Response Time
 - Throughput
- **Interesting questions can be asked**
 - What is the slowest layer?
 - Which kind of requests are slow?
 - Who is sending the slowest requests?
 - Who is actually using my service?
- **Auditing Extension + Analytics Stack is what you need!**

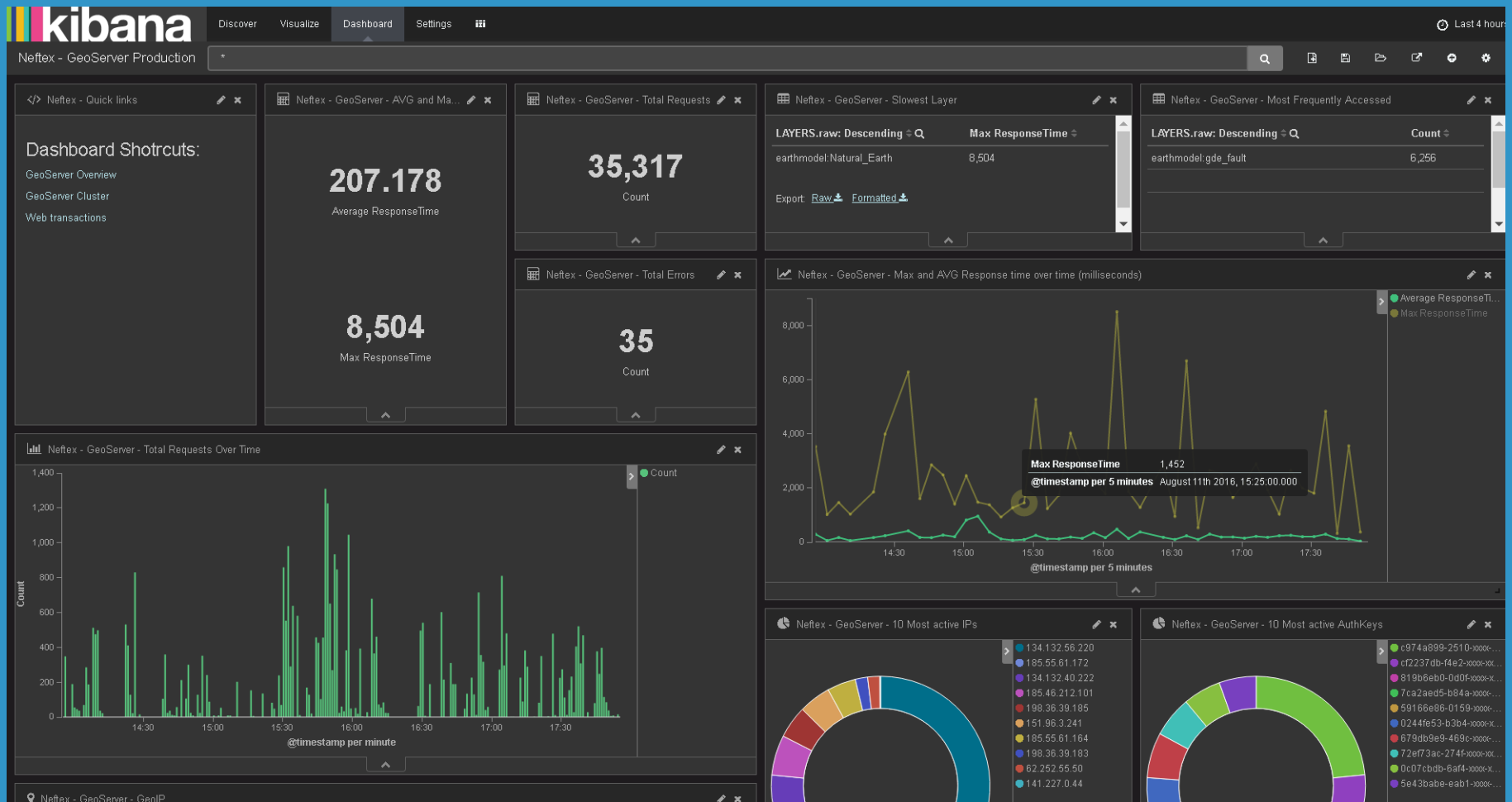




When in production



- Metering



GeoSolutions
 your one-stop-shop for geospatial open source software




FOSS4G 2017, Boston
August 14th-19th 2017



When in production



- Document the entire infrastructure
- Monitor every bit
- Use alerts and actions to be proactive
- Keep calm and take snapshots before taking actions
- Is there any magic?
 - Well knowing your components inside out helps a lot 😊



That's all folks!



Questions?

info@geo-solutions.it

