GeoServer
Past, Present and Future

Andrea Aime
TOPP
Outline

• Now and then... a lighting comparison
  – 9 releases
  – 1.3.4 vs 1.6.0 beta3 at a glance
• One year of progress
  – lots of new features
  – fixes and speed improvements all over the place
• The future
  – see what's cooking in developers minds
Evolution at a glance
One year of releases

- **SEPT 06**
  - **1.3.4**
  - minor bugfixes

- **OCT 06**
  - **1.4.0**
  - modular arch.
  - many little improv.

- **NOV 06**
  - **1.4.1**
  - fixes and improv.
  - Oracle speedup

- **DEC 06**

- **JAN 07**

- **FEB 07**

- **MAR 07**
  - **1.5.0**
  - WCS
  - rasters in WMS

- **APR 07**

- **MAY 07**

- **JUN 07**

- **JUL 07**
  - **1.6.0 beta2**
  - WFS 1.1
  - Versioning WFS
  - faster shapefile rend.
  - improved conn. pool
  - security subsystem

- **AUG 07**

- **SEPT 07**
  - **1.5.2 & 1.5.3**
  - paletted PNG
  - improv. extended char support
  - improv. GetLevedGraphics
  - Oracle WFS-T improvements
  - GetFeatureInfo templates

- **1.6.0 beta3**
  - faster WMS
  - GeoJSON
  - cascaded WFS
1.3.4 vs 1.6.0 beta3

- We'll compare:
  - GeoServer 1.3.4, which saw the light *right after FOSS4G 2006* (with minor fixes)
  - GeoServer 1.6.0 beta3, which saw the light *one week ago*

- Comparison as a data gateway
- Three benchmarks
GeoServer

Shapefile
GML
VPF

PostGIS
DB2
Oracle
MySql

ArcSDE
WFS

Vector files

WFS 1.0

WMS 1.1

1.3.4

DBMS

Servers

Crude vector data

Styled maps

PNG, GIF
JPEG
SVG, PDF
KML/KMZ

Shapefile.zip
GML2
GML2.gz
Postgis rendering

<table>
<thead>
<tr>
<th>Threads</th>
<th>GS 1.3.4</th>
<th>GS 1.6.0beta3</th>
<th>GS 1.6.0beta3 n.a.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>221</td>
<td>195</td>
<td>109</td>
</tr>
<tr>
<td>2</td>
<td>357</td>
<td>349</td>
<td>138</td>
</tr>
<tr>
<td>4</td>
<td>805</td>
<td>735</td>
<td>195</td>
</tr>
<tr>
<td>8</td>
<td>1697</td>
<td>1469</td>
<td>462</td>
</tr>
</tbody>
</table>

10 requests per thread
1000 out of 10,000 polygons rendered
n.a.: non antialiased
Shapefile rendering

10 requests per thread
1000 out of 10,000 polygons rendered
n.a.: non antialiased

<table>
<thead>
<tr>
<th>Threads</th>
<th>GS 1.3.4</th>
<th>GS 1.6.0beta3</th>
<th>GS 1.6.0beta3 n.a.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>224</td>
<td>159</td>
<td>73</td>
</tr>
<tr>
<td>2</td>
<td>459</td>
<td>281</td>
<td>101</td>
</tr>
<tr>
<td>4</td>
<td>729</td>
<td>608</td>
<td>123</td>
</tr>
<tr>
<td>8</td>
<td>1544</td>
<td>1258</td>
<td>294</td>
</tr>
</tbody>
</table>
WFS encoding

10 requests per thread
Encode in GML2 all features in the sample layer topp:states

<table>
<thead>
<tr>
<th>Threads</th>
<th>GS 1.3.4</th>
<th>GS 1.6.0beta3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>117</td>
<td>109</td>
</tr>
<tr>
<td>2</td>
<td>122</td>
<td>110</td>
</tr>
<tr>
<td>4</td>
<td>207</td>
<td>176</td>
</tr>
<tr>
<td>8</td>
<td>469</td>
<td>393</td>
</tr>
</tbody>
</table>
One year of new features
WCS, WMS and raster data

• 1.5.0 saw the introduction of raster support, both as WMS and as WCS
• Thanks GeoSolutions! (www.geosolutions.it)

• GeoTiff, Image + world file, ArcGrid, Gtopo30
• Large images support
  – inner tiled GeoTiffs with overviews
  – image mosaics
  – image pyramids
WMS raster integration

ArcGrid precipitation raster + raster symbolizer

Bluemarble TNG portion, 2GB inner tiled GeoTIff
Web Coverage Service

- The WFS of raster world
- Extract raster data from available sources
- Certified OGC compliant (WCS 1.0)
Mashups everywhere

- New support for GeoRSS and GeoJSON, as well as Google Maps WMS integration.
- Improved Google Earth support
- Want more? Attend the next presentation
Color me ${templated}

• User customization of various outputs:
  – KML/KMZ (Google Earth & Maps)
  – GeoRSS (Yahoo! Maps, Virtual Earth)
  – WMS GetFeatureInfo (OGC clients)

• Template support courtesy of Freemarker, (www.freemarker.org)

• Want to see more? Attend the next presentation!
Paletted images

• The smaller the file, the faster the download
• Using 256 colors (or less) png's makes a world of difference
• Which palette?
  – compute one on the fly by image inspection -> `image/png8` and `image/gif`
  – user provide one -> `palette=name` (“safe”, palette by example, .pal file)
  – derive it from the SLD style -> `format_options=antialias:none`
Road layer

- **png**, 27ms, 25kb (TT 177ms)
- **png8**, 53ms, 10kb (TT 143ms)
- Safe pal., 27ms, 8kb (TT 109ms)
- No ant., 17ms, 3kb (TT 79ms)
Raster data

- png, 40ms, 28kb (202ms)
- png8, 49ms, 20kb (179ms)
- safe pal., 28s, 5kb (98ms)
- jpeg, 23ms, 6kb (TT 97ms)
Web Feature Service 1.1

- The new WFS spec:
  - GML3
  - reprojection
- Soon to become an ISO standard (with a few amendments that will make it WFS 1.2)
- GeoServer is the reference implementation for WFS 1.1 too
Versioning WFS

• Think WFS + a version control system (CVS, SVN):
  – access to a specific revision
  – rollback
  – diff
  – user identification, commit log, log access

• Yet, backwards compatible with standard WFS: transparent versioning

• Not a spec, but an experiment and a proposal
• Multi-user, multi role
• Remember login cookies
• HTTP basic authentication on OGC services

• At the moment, a promising prototype configured with two clear text files:

users.properties
#user=password,role1,...,roleN
admin=geoserver,ROLE_ADMINISTRATOR
wfst=wfst,ROLE_WFS_READ,ROLE_WFS_WRITE
wfs=wfs,ROLE_WFS_READ

services.properties
#service.request=role1,...,roleN
wfs.GetFeature=ROLE_WFS_READ
wfs.Transaction=ROLE_WFS_WRITE
The future
KML / Google Earth

• 'Extrudes' - 3d look from your GIS data in Google Earth from attributes.
• More efficient streaming of large datasets to Google Earth
  – Raster super-overlays with Java tile caching
  – Investigation of Vector region streaming
• Participating in OWS-5 testbed on Agile Geography to help define what the next version of KML will be.
Raster data

- GeoServer is going to become the WCS 1.1 reference implementation
- Native multi dimensional raster handling (NetCDF, HDF)
- GDAL integration, so access to ECW, MrSid, JPEG 2000 and other formats
Other exciting stuff

• REST interfaces:
  – Remote administration
  – Atom publishing protocol (FeatureServer like)

• Visual SLD editor

• Integrated tile cache
  – Ease of configuration
  – WFS-T integration, expired tiles for modified data

• Much, much more, see the Bonus Tracks
Question time!
(aaime@openplans.org)
Bonus tracks

Good stuff... that did not fit the 25 minutes time limit
Standard template

```
<table class="featureInfo">
  <caption class="featureInfo">${type.name}</caption>
  <tr>
    <#list type.attributes as attribute>
    <#if !attribute.isGeometry>
      <th>${attribute.name}</th>
    </#if>
    </#list>
  </tr>
  <#assign odd = false>
  <#list features as feature>
  <#if odd>
    <tr class="odd">
  </#else>
    <tr>
  </#if>
  <#assign odd = !odd>
  <#list feature.attributes as att>
  <#if !attribute.isGeometry>
    <td>${att.value}</td>
  </#if>
  </#list>
  </tr>
  </#list>
</table>
```

Table header

Alternate row background control

Attribute list
Custom templates

List oriented, but can still be applied to every feature type

Feature type specific, knows about attribute name and meaning
Paletted images extra
Some benchmarks

- 300x200 images
- Parameters
  - `image/png`: full color, 24bits per pixel png
  - `image/png8`: 256colors, 8bits png, rendered in full color and reduced to 256 in post processing
  - `image/png&palette=safe`: 256colors, 8bit png, rendered in full color and reduced to a known palette in post processing
  - `image/png&format_options=antialias:none`: palette computed from SLD, rendering in 256 colors directly, no antialiasing
Some benchmarks (cont.)

• Times measured with **ab** (ApacheBench) and 30 requests
• Total Time considers a 2Mbit/s connection and a 50ms latency
Beware the safe palette

- png, 25ms, 11kb (TT 119ms)
- png8, 38ms, 6kb (TT 112ms)
- safe pal., 20ms, 4kb (TT 86ms)
- no ant., 12ms, 3kb (TT 74ms)
Multilayer, labelled map

- **png**
  - 139ms, 95kb (516ms)

- **png8**
  - 143ms, 24kb (289ms)

- **safe pal.**
  - 127s, 22kb (TT 265ms)

- **no ant.**
  - 108ms, 15kb (TT 258ms)
Image sampler conclusions

• On a real internet connection, png8 is always faster than png
• Using a fixed palette is even better, but better provide a custom one, specific for the custom map
• Non antialiased images are real fast for simple overlay layers, but quality is low
• In raster background serving nothing beats JPEG.
WFS-V sample
Step 1, doing a transaction

```
<wfs:Transaction service="WFSV" version="1.0.0"
...
handle="Updating Signature rock label">
  <wfs:Insert>
    <topp:archsites>
      <topp:cat>2</topp:cat>
      <topp:str1>Alien crash site</topp:str1>
      <topp:the_geom>
        <gml:Point srsName="http://www.opengis.net/gml/srs/epsg.xml#26713">
          <gml:coordinates decimal="." cs="," ts=" ">
            604000,4930000
          </gml:coordinates>
        </gml:Point>
      </topp:the_geom>
    </topp:archsites>
  </wfs:Insert>

  <wfs:Update typeName="topp:archsites">
    <wfs:Property>
      <wfs:Name>str1</wfs:Name>
      <wfs:Value>Signature Rock, updated</wfs:Value>
    </wfs:Property>
    <ogc:Filter>
      <ogc:FeatureId fid="archsites.1" />
    </ogc:Filter>
  </wfs:Update>

  <wfs:Delete typeName="topp:archsites">
    <ogc:Filter>
      <ogc:FeatureId fid="archsites.2" />
    </ogc:Filter>
  </wfs:Delete>
</wfs:Transaction>
```
Step 2, visual compare

http://.../geoserver/wms?request=GetMap
&HEIGHT=400&WIDTH=600
&LAYERS=topp:archsites&...
&featureVersion=1

http://localhost:8080/geoserver/wms?
request=GetMap
&HEIGHT=400&WIDTH=600
&LAYERS=topp:archsites&...
Step 3, get a diff

```
<wfsv:GetDiff
service="WFSV" version="1.1.0"
outputFormat="HTML"
...>
<wfsv:DifferenceQuery
typeName="topp:archsites"
fromFeatureVersion="1"/>
</wfsv:GetDiff>
```
Step 4, roll back and log

```
<wfs:Transaction
service="WFSV"
    handle="Rolling back
           previous changes">
  <wfs:Rollback
    safeToIgnore="false"
    vendorId="TOPP"
    typeName="archsites"
    toFeatureVersion="1"/>
</wfs:Transaction>

<?xml version="1.0" encoding="UTF-8"?>
<wfs:TransactionResponse
    version="1.1.0">
  <wfs:TransactionSummary>
    <wfs:totalInserted>1</wfs:totalInserted>
    <wfs:totalUpdated>1</wfs:totalUpdated>
    <wfs:totalDeleted>1</wfs:totalDeleted>
  </wfs:TransactionSummary>
  <wfs:TransactionResults />
  <wfs:InsertResults>
    <wfs:Feature>
      <ogc:FeatureId fid="archsites.2" />
    </wfs:Feature>
  </wfs:InsertResults>
</wfs:TransactionResponse>

<wfs:GetLog service="WFSV"
    outputFormat="HTML">
  <wfs:DifferenceQuery
    typeName="topp:archsites"
    fromFeatureVersion="1"
    toFeatureVersion="3"/>
</wfs:GetLog>
```

<table>
<thead>
<tr>
<th>Revision</th>
<th>Author</th>
<th>Date</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>anonymous</td>
<td>20/09/07 9.39</td>
<td>Rolling back previous changes</td>
</tr>
<tr>
<td>2</td>
<td>anonymous</td>
<td>19/09/07 17.06</td>
<td>Updating Signature rock label</td>
</tr>
</tbody>
</table>
Future extras
GML Complex features

- Support for complex features, that is a feature with:
  - nested subfeatures
  - collection attributes
  - associations

- This will open the road for sensor observations and other non flat GML data

- Mapping from database schema to well known GML schema
REST interfaces

- Remote configuration of GeoServer Layers
- User and permission management

- Atom Publishing Protocol implementation for editing geodata
- WFS alternative
- Hook up to versioning backend
Integrated tile cache

- Integrated Java TileCache
- Summer of Code Project built JTileCache
- Ship with GeoServer with seamless integration and configuration
- Hook up to transactions, with conditional HTTP, for live edits with cached data (invalidate only affected tiles)
- Investigate caching vector outputs
• Visual SLD editor
• Ajax based environment to style maps
• Thematic mapping (prototype is complete)
• Graphical User Interface to make maps look good
• Scale centered UI for web maps.
New and improved subsystems

- New configuration subsystem, more amenable to cluster, and quicker to develop with for developers
- Remote access to configuration (read/write)
- New modular user interface, more options, quicker setup
- Improved security subsystem (from prototype to production grade)