A Comparison of ArcIMS to MapServer

Brock Anderson
Refractions Research
Mapserver versus ArcIMS

- Victoria, BC, Canada
- Spatial Systems Consulting
- Open Source Software
- PostGIS
- uDig / GeoTools
• BC’s Ministry of Sustainable Resource Management (Information Management Branch)
  
  – Use mostly ESRI products in their mapping infrastructure.
  – Were frustrated with ArcIMS’s administrative neediness.
  – Were looking for WMS alternatives to ArcIMS. We suggested MapServer.
We prepared a detailed test plan to compare ArcIMS and MapServer as WMSs.

The tests covered:
- Ease of administration
- WMS 1.1.1 standard compliance
- Interoperability with other software
- Performance

* Ease of administration
  - KEY TEST: time to add and reload services

* Interoperability with other software
  - KEY TEST: supports ArcSDE 8.3 and 9.x?

* WMS Standard compliance
  - KEY TEST: OGC WMS CITE

* Performance (with ArcSDE)
  - KEY TEST: feature density
  - KEY TEST: feature complexity
  - KEY TEST: image output format
  - KEY TEST: concurrency
  - KEY TEST: reprojection
  - KEY TEST: throughput "under regular operating conditions"
Jmeter is open source software designed to load test functional behavior and measure performance.
Test Preparations

- Created synthetic data to be used for certain performance tests.

- Made a JMeter extension to simulate a diversity of GetMap requests.

- Created ArcIMS AXL files and MapServer map files.

- Synthetic data used for feature density and feature complexity test.
- JMeter extension used to generate WMS requests with random bboxes.
- Started with AXL files, and transformed them into .map files
Our client had an “optimized” .axl file containing their provincial basemap data.

We used that .axl file to create a .map file from.
• Early performance tests showed ArcIMS outperformed MapServer

• An initial test to convince ourselves that MapServer was comparable to ArcIMS
- We requested the same map each time: 4 layers (including points, lines and polygons)
- Each level of concurrency run for 10 minutes.

• Mention that 1) all graphs have at least 30 samples per point. 2) All tests done against ArcSDE.
Profiling revealed two main bottlenecks:

- 1 to 2 seconds of ArcSDE connection overhead per GetMap request

- Additional overhead extracting features from ArcSDE
Minimizing the Connection Overhead

- Persistent database connections would nearly eliminate ArcSDE connection overhead.

- As a CGI program, MapServer had no means to support persistent connections.

- Added FastCGI support. Thanks Frank!

- Updated the ArcSDE module to utilize persistent connections. Thanks Howard!

- Frank Warmerdam added FastCGI support to MapServer, and he created a connection pooling API
- Howard Butler updated the ArcSDE module to utilize connection pooling
Mapserver versus ArcIMS

Using MapServer with FastCGI

1. Compile MapServer with FastCGI support.
2. Configure your web server with a FastCGI module.
3. Update your map files:

```
LAYER
  PROCESSING "CLOSE_CONNECTION=DEFER"
  # all other layer settings here...
END #LAYER
```

* Persistent connections are most useful for data sources with large connection overhead, such as ArcSDE.
Other Enhancements

- Recall, connection overhead was not the only slowdown.

- We also improved the MapServer code which pulled features from ArcSDE.
  
  - This involved experimenting with ESRI’s ArcSDE C API.
  - We discovered which operations were costly, and cut down on their use.

*Performance is best for the default version of ArcSDE layers.*
The earlier performance test was run again, this time with FastCGI and the other performance improvements.
MapServer now outperforms ArcIMS under MSRM’s “regular operating conditions.”

- The higher the better on this graph only
- Difference between concurrency test and this test:
  - This test uses random requests for 10 min.
  - This test doesn't hit the server with a sudden burst of requests. They are ramped up.
MapServer is slightly faster to return GIF and PNG images. ArcIMS is faster for JPG.
MapServer reprojects faster than ArcIMS.

Reprojection vs. Response Time

- BC Albers (NAD83)
- Lat-Long (NAD 83)
- UTM10 (NAD 83)

Output Projection

Average Response Time (ms)

ArcIMS
MapServer
• Feature complexity affects both servers almost equally (from ArcSDE).
MapServer versus ArcIMS

- MapServer extracts features from SDE slightly faster than ArcIMS does.

Actually, this test demonstrates that MapServer (extracts features from SDE) + (draws the map) faster
MapServer versus ArcIMS

- MapServer passed all 83 WMS CITE tests.
- ArcIMS (w/ WMS connector) passed 71 of 83 WMS CITE tests.
  - ArcIMS failed 12 tests because:
    - Wrong MIME type for some responses.
    - Wrong "exception code" in some exceptions.

• MIME type of responses is typically expected to be "application/vnd.ogv.se_xml"
Mapserver versus ArcIMS

Ease of Administration

• MapServer administration benefits
  - No need to reload services (when service files change)
  - MapServer restarts faster (as fast as the web server)

• ArcIMS administration benefits
  - More granular control over log levels
Interoperability and Support

- Both WMSs support:
  - ArcSDE 8.3 and ArcSDE 9.0
  - Styled Layer Descriptor (SLD)

I don’t know how fully either server supports SLD, but our testing showed that both support these basic features:
- Selecting and styling based on a attribute value
- Selecting and styling based on spatial location (within a bounding box)
### Conclusions

- MapServer is easier to administer than ArcIMS.
- MapServer is more WMS standard compliant than ArcIMS.
- MapServer matches or surpasses ArcIMS in most performance tests.

* (MSRM adopted MapServer for their COINPacific web mapping application)
Contact me:

Brock Anderson
banders@refractions.net

Refractions Research
www.refractions.net
(250) 383-3022