Optimizing ArcIMS Services

By Sarah Osborne and Martin Fleming, ESRI

Editor's note: *Optimizing the content of services for an ArcIMS Web site will reduce the amount of work needed to create map images. It is one of the most effective ways of improving site performance. This article describes a variety of optimizing strategies that can be implemented with only a modest effort. It is assumed that the reader is familiar with the structure and basic administration of ArcIMS.*

Making small enhancements to map configuration files can dramatically improve Web site performance by reducing response time and increasing the throughput of the ArcIMS Server. The total response time for an ArcIMS Server reflects

- The time the request is queued
- The time needed to fetch data
- Rendering time in the Spatial Server
- The time needed to send the response back to the client

For Intranet applications, rendering time in the Spatial Server has the greatest effect on overall response time. For Internet applications, bandwidth to the client must also be considered. These simple steps will shave seconds off rendering time and make the entire site more responsive.

Measuring Map Configuration File Performance

Suppose you have an ArcIMS Image Service that takes five seconds to return a response. However, based on users' expectations, response time of two seconds or less is needed. How can you reduce the generation time by three or more seconds? Start looking at the Spatial Server log files. This will give you an idea of how much time is spent fetching data and rendering features. If your requests routinely take longer than two seconds to process, the service would likely benefit from some tuning.

Map configuration files tell the Spatial Server which data to publish and how that data should be symbolized and labeled. Analyzing Spatial Server log files is the best way to start improving map configuration file performance. These files record a wealth of information about the activities required to service a request. The time spent finding data, retrieving data, and processing features is logged for each layer in the service. If one layer takes significantly longer to render than other layers, this could indicate a bottleneck that could be removed through optimization.

Take a look at Figure 1. This log file excerpt shows the types of logged information that measure performance for an Image Server when verbose logging is turned on. For Image Services, the log file records the time a request was received, the request type (e.g., GET_IMAGE), information about each layer, and both the ArcXML request and response. Log files provide a good place to compare changes in response time before and after making those changes to evaluate their effectiveness.

Tuning Strategies

The amount of time it takes the Spatial Server to render features is directly proportional to the number of vertices and line segments used to represent those features. A map that shows less data will render more quickly. The following strategies will help determine how to minimize rendering time by limiting the quantity and detail of the map data while supporting the purpose of the service.

The high level of detail used when plotting paper maps is typically lost when the data is converted to an image for display using a Web browser. A standard video display is usually limited to 96 pixels per inch, a resolution that is much lower than the 300 to 600 dots per inch used by printers. Consequently, video display cannot show as much information. Assessing the detail at any given scale is particularly important for services that cover a broad range of scales (e.g., national to local views) and a broad set of geographic areas (e.g., urban and rural areas).

When reviewing log files, look closely at the data search and data retrieval times along with the number of features processed per layer. Identify any layers in the service for which the data retrieval times are particularly slow (i.e., retrieval time exceeding 0.10 second for a layer). Determine if any layers in a request return an excessive number of features (i.e., more than 100 features per layer). For these layers, consider implementing one or more of the following solutions:

- Set scale dependencies.
- Use generalized features.
- Use geoprocessing to reduce the number of features.
- Use ArcSDE group layers.
- Limit the number of labels.
- Review symbol rendering.

Set Scale Dependencies

The four maps in Figure 2 show public land survey system polygons displayed at various scales. As the number of features displayed on each map increases, processing time increases dramatically and

Log File Terms

Search Time is measured from the moment the Spatial Server first sends a request for data until the time the first feature is received.

Retrieval Time is measured from the time the first feature is received until the last feature is received.

Blocks Retrieved refers to a buffer inside the Spatial Server to hold image data. If a query for image data returns zero blocks, the requested extents did not intersect with the extents of any image data.

Total Processing Time denotes the time elapsed during the rendering phase of map creation. It is measured from the end of the renderer setup phase until the map image is complete, just before the image is compressed and written from memory to disk.

Total Request Time is the overall elapsed time of the Spatial Server instance to create a map including AXL parse time, renderer setup, search and retrieval, rendering, compression, and output to disk.

```
Image Server log file – GET_IMAGE request
All requests start with a 'Begin Request' notification.
[Jan 15, 2004 1:59:42 PM][3032 2216 INFO1] Begin Request

The request received from the ArcIMS Application Server.
[Jan 15, 2004 1:59:42 PM][3032 2216 INFO3] REQUEST: <GET_IMAGE>
<PROPERTIES>
<ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
<IMAGESIZE width="800" height="600"/>
</PROPERTIES>
</GET_IMAGE>

Request type (GET_IMAGE or GET_RASTER_INFO) and the name of the service
(world).
[Jan 15, 2004 1:59:42 PM][3032 2216 INFO1] GET_IMAGE: world

Time needed to parse the request in seconds.
[Jan 15, 2004 1:59:42 PM][3032 2216 INFO2] AXL Parse Time: 0.000001s

Time needed to allocate space for drawing the image.
[Jan 15, 2004 1:59:42 PM][3032 2216 INFO2] RENDERER SETUP: 0.000000s

Name of the layer being processed.
[Jan 15, 2004 1:59:42 PM][3032 2216 INFO2] FEATURE LAYER: countries

Time to create and send a query to the data source.
[Jan 15, 2004 1:59:42 PM][3032 2216 INFO2] DATA SEARCH TIME: 0.016000s

Number of features processed for the layer.
[Jan 15, 2004 1:59:42 PM][3032 2216 INFO2] SR FEATURES PROCESSED: 12

Time to process and fetch features.
[Jan 15, 2004 1:59:42 PM][3032 2216 INFO2] DATA RETRIEVAL TIME:
0.063000s

Time to draw all labels.
[Jan 15, 2004 1:59:42 PM][3032 2216 INFO2] LABEL ENGINE TIME: 0.000000s

Total time after the request has been parsed until labels are drawn.
Note that this total time is not usually exactly equal to the sum of
renderer setup, data search time, data retrieval time, and label engine
time.
[Jan 15, 2004 1:59:42 PM][3032 2216 INFO2] TOTAL PROCESSING TIME:
0.079000s

Time to generate the image.
[Jan 15, 2004 1:59:43 PM][3032 2216 INFO2] OUTPUT TIME: 0.078000s

Response sent to the Application Server.
[Jan 15, 2004 1:59:43 PM][3032 2216 INFO3]
RESPONSE: <?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<RESPONSE>
<IMAGE>
<ENVELOPE minx="-180" miny="-135" maxx="180" maxy="135" />
<OUTPUT url="http://mymachine/output/canada_MYMACHINE3032221613.jpg" />
</IMAGE>
</RESPONSE>
</ARCXML>

Time spent for all processing starting with 'Begin Request' and ending
with 'End Request'.
[Jan 15, 2004 1:59:43 PM][3032 2216 INFO1] Total Request Time:
0.188000s

Request is fully processed.
[Jan 15, 2004 1:59:43 PM][3032 2216 INFO1] End Request
```

Figure 1: Logged information that measures performance for an Image Server

the map content becomes more cluttered and less informative.

If a layer is displayed at a scale that contains too many features, set it to a larger scale. For example, make detailed streets visible at 1:25,000 rather than 1:40,000. In practice, setting the scale often involves making compromises between content and performance, particularly when the service covers a diverse area. Adjusting scale dependencies can be a quick fix for layers that cause performance problems.

Use Generalized Data for a Layer

If a layer contains too many features for a given scale but it is important to display this layer, consider using a modified layer that either contains fewer features or simplified features. (Note that data generalization is accomplished in ArcGIS, ArcInfo, or ArcView.) The level of detail originally shown may be unnecessary once the map is converted to a low-resolution map image. For example, when displaying a river system at disparate scales, consider using multiple data layers that have been developed with the appropriate level of generalization for various scales such as 1:100,000, 1:1,000,000, and 1:10,000,000.

Continued on page 56

Optimizing ArcIMS Services

Continued from page 55

Use Geoprocessing to Reduce Features

Grouping features reduces the number of features that must be read. However, this strategy can also involve a trade-off. For example, a road layer used for geocoding divides individual roads into many short segments with address attributes. Grouping these features by road name reduces the number of features read but eliminates the ability to geocode the layer once the address information is merged.

Use ArcSDE Group Layers

ArcSDE can be used to create a grouped feature class. ArcSDE treats a grouped feature class no differently than any other feature class. Grouping features together either by attribute or spatially reduces the number of rows in the database. Grouped feature classes require less space and can be retrieved more quickly.

Don't Use Too Many Labels

Although ArcIMS allows labeling of any layer in a map configuration file, it is a good idea to label only one or, at most, a few layers. Labeling many layers can result in longer response times and requires the Spatial Server to do more work. As labels are crowded, the time it takes for the Spatial Server to place them will increase. In addition, too many labels can be confusing to the map viewer. Remember that labels must be legible at screen resolution.

Special labeling effects, such as glowing, drop shadows, background colors, and antialiasing in particular, will impact total response time. Although these effects increase the total labeling time by a small fraction, this overhead should not be ignored. Get an estimate of how long labeling takes by reviewing the label engine time shown in the log files.

Label scale factors can be set separately from other features. Labels do not need to display the entire time a layer is visible. Instead, turn on label visibility at a larger scale than feature visibility to decrease the amount of work the Spatial Server needs to do for every request.

Review Symbol Rendering

Choose symbology that is appropriate for the scale. At small scales, complex symbology is distracting and can slow down image output time. Each part of a multipart symbol is rendered separately, so a four-part symbol is drawn four times by the Spatial Server. Using image markers and raster fills can increase processing time if many images must be placed on the map. Also, antialiasing a symbol can increase processing time. Finally, if possible, use unique or graduated symbols at only larger scales. Extra processing time is required because the Spatial Server must access attribute data to render each feature based on its attribute value.

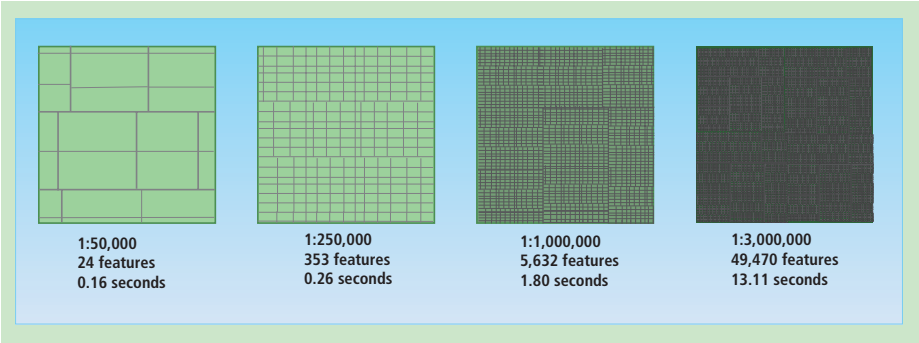

Figure 2: The effect of the number of features on response time when retrieving four maps created at various scales

Be Careful With Raster Data

ArcIMS offers several ways to include raster data in an Image Service: referencing rasters individually, using image directories or catalogs, or using ArcSDE. Although the service may look the same, some methods may incur a performance penalty.

Avoid Services With Hundreds of Layers

Adding rasters individually can cause performance problems because the Spatial Server must parse the reference for each image from the map configuration file. If many images must be added to a service and these images reside in the same directory, use an ArcIMS image directory, which allows all images in a directory to be referenced once in the map configuration file. Try to keep the number of layers to a minimum.

Use TIFF Files Wisely

Using TIFF images can severely impact performance if they are displayed at a scale that requires high levels of resampling. Zoom out to a smaller scale, and the Spatial Server must read large amounts of data from the TIFF files and resample it to the coarser resolution needed for output images. This means additional overhead in terms of memory and processing time. Storing data in ArcSDE is one alternative to TIFF files. ArcSDE uses pyramid layers, which store resampled images in the database and reduce the I/O and processing needed for a given map.

Limit the Data Retrieved

The SPATIALQUERY element can be used to set constraints on the data retrieved in several ways.

Reduce the Number of Accessible Attribute Fields

Some data fields in a layer may not be of interest to users of an ArcIMS site. Removing unneeded fields can result in faster Spatial Server performance and smaller response strings returned to the client.

This will also remove unnecessary clutter from the application. Unused attributes in database tables cause the hard drive to read and discard data. Even if only a fraction of the information contained in a database block is needed, the database and operating systems will read the whole block. Deleting unused attributes can reduce the number of blocks read and reduce the time spent waiting on the hard drive.

If it is not possible or practical to physically remove unneeded fields from the database, limit which fields are available by using the SPATIALQUERY subfields attribute. Listing only those fields that should be available for the service will cause the Spatial Server to limit requests to those fields.

Limit the Number of Features That Can Be Requested

Limit the number of features, on a per layer basis, that can be returned in any one request by using the SPATIALQUERY featurelimit attribute.

```
      <LAYER type="featureclass" name="Cities" visible="true" id="4">
   <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
<SPATIALQUERY subfields="name population" featurelimit="25" searchorder="
attributefirst"/>
   <SIMPLERENDERER>
     <SIMPLEMARKERSYMBOL color="102,0,102" width="8.0" />
   </SIMPLERENDERER>
      </LAYER>
```

Figure 3: Using SPATIALQUERY to specify search order

Use the Best Search Order for Both Data and Application

When using ArcSDE data layers, the Spatial Server can search data two ways—by attribute or spatially. Specifying whether to search data by attribute first or spatially first can affect performance. In Figure 3, SPATIALQUERY has been added to a cities layer. The fields available for query are limited to the city name and population. A maximum of 25 features will be returned for any request. The search order has been set to attributes first.

Additional Considerations

These strategies will also enhance performance by limiting rendering overhead.

Use the Acetate Layer Judiciously

The acetate layer, designed for only a few features—such as a north arrow, scale bar, copyright statement, and perhaps a couple of graphics—should not be used for adding large amounts of data. Performance can slow considerably if too many features are included on the acetate layer.

Minimize Changing Projections on the Fly

Although changing projections on the fly can affect performance, this effect will be limited if scale factors have been used to reduce the number of features being rendered.

Use an Appropriate Output Image Format for the Service

Using an image format that is not optimized for the type of data being served can adversely affect performance. Vector data can usually be served in GIF or PNG8 format. Rasters should be served in JPEG or PNG24 format. However, if transparency is required, the JPEG format does not support transparency and Internet Explorer does not support PNG24 transparency. Making an informed decision about the output image format can result in smaller output image file sizes that save bandwidth and decrease Spatial Server processing time.

Don't Allow Users to Request Very Large Images

The size of an output map image has an impact on performance. Use the properties for a service to limit the amount of memory used by a Virtual Server instance to render a map. The default is an image size approximately 1,024 x 1,024 pixels. If the application normally uses smaller images, lower the memory limit to prevent other applications from requesting a larger map.

Summary

These strategies will increase performance without requiring additional hardware or software. These minor changes can have a big effect on the overall performance of an ArcIMS Web site. Changes, such as setting scale factors, generalizing features, and limiting labeling, may also produce map images that are less cluttered and more readable online for map viewers.

For Additional Information

For more information on ArcXML and map configuration files, see the *ArcXML Programmer's Reference Guide*. This guide is available online at support.esri.com. ArcIMS server administrators who want to understand the workings, motivations, and performance considerations of ArcIMS technology may also benefit from the ESRI instructor-led course, *ArcIMS Administration*. Visit www.esri.com/training for a detailed course description and information on registration and other courses. The ESRI Virtual Campus also offers ArcIMS courses online. Visit campus.esri.com to read course descriptions or register.