ArcGIS Server Administration, Configuration, Tuning, and Optimization

Michele Lundeen-ESRI
Topical Outline

- Preparing GIS documents
- Publishing GIS services
- Creating and Configuring GIS applications
- Administration & System Monitoring
- Deployment Architectures
Preparing GIS Documents

**General Overview**

- Understand application requirements
- Design specifically for server deployment
  - Maps
    - “Web 2.0”
    - Cached – “basemap” data
    - Dynamic – “operational” data
  - Geoprocessing Models
    - Preprocessing
    - Limit inputs
  - Mobile Applications
    - Devices
    - Environment
  - Geocoding
    - Locator locations
Web Maps 1.0

Common pitfalls

- Too many data layers
- Toggling every layer on/
- Poor cartography
- No scale dependency
- Slow dynamic drawing
Web Maps 2.0
Change your approach

• Only include layers that support the business need

• Logical grouping of layers

• Small number of layers to turn on and off

• Choose the best option for publishing the map
  – Cached tiles
  – Dynamic layers
  – Client-side graphics
Organize data into logical groupings

**Basemaps**

- Geographic frame of reference
- Contain static vector and raster data
- Reusable in multiple applications

**Operational Layers**

- Show a focused item of interest
- Support functionality of the application
- Displayed on top of base map
Some ArcGIS Server examples

• Orange County Property Appraiser Map

• Solar Boston

• City of Greeley Property Information Map
Case study: Google Maps
### Google Maps base maps

<table>
<thead>
<tr>
<th>&quot;Map&quot;</th>
<th>&quot;Terrain&quot;</th>
<th>&quot;Satellite&quot;</th>
</tr>
</thead>
</table>
| • Highways  
• Streets  
• Ferries  
• Railroads  
• Transit centers  
• Cities  
• Parks  
• Military reservations  
• Municipal boundaries  
• Lakes  
• Rivers  
• Golf courses  
• Hospitals  
• Shopping centers  
• Airports  
• Colleges  
• Cemeteries  
• Amusement parks | • Shaded relief  
• Vegetation  
• Highways  
• Streets  
• Cities  
• Parks  
• Military reservations  
• Municipal boundaries  
• Lakes  
• Rivers  
• Golf courses  
• Hospitals  
• Shopping centers  
• Airports  
• Colleges  
• Cemeteries  
• Amusement parks | • Imagery  

Preparing GIS Documents

General Overview

• Understand application requirements
• Design specifically for server deployment
  – Maps
    • “Web 2.0”
    • Cached – “basemap” data
    • Dynamic – “operational” data

  – Geoprocessing Models
    • Preprocessing
    • Limit inputs

  – Mobile Applications
    • Devices
    • Environment

  – Geocoding
    • Locator locations
Benefits of Map Cache

- Best performance
- Improved scalability
- More options for cartography
Working with Map Cache

Best Practices

- Chose appropriate image output type
  - Raster cache use JPEG
  - Vector cache use PNG

- Match scale dependency levels with cache levels

- Same projection for all services
  - Coordinate if you plan to share cache or do mash-ups
  - Avoid projection on the fly

- Create a test cache using a small area
Resources for Building Map Cache

• **SERUG session**
  - Designing, Deploying, and Using Cached Map Services  Wed 8:30am

• **Virtual Campus Seminar**
  - *Implementing and Optimizing ArcGIS Server Map Caches*

• **ArcGIS Documentation: Caching Services Topic**

• **ArcGIS Server Blog, Map Cache Tag**
Preparing GIS Documents

Dynamic Map Services: Feature Rendering

• Points
  – Use single layer Simple or Character markers for best performance
  – Use EMF instead of bitmaps
  – Use Integer (vs. character) fields for symbol values
  – Avoid halos (MSD review), complex shapes, masking

• Line & Polygons
  – Use ESRI_Optimized style (for MXD Service only)
  – Avoid cartographic lines (also includes polygon outline!)
Preparing GIS Documents

Dynamic Map Services: Feature Rendering

- Avoid Projecting on the fly
- Avoid cross db joins
- Avoid wavelet compression-based raster types (MrSid, JPEG2000)
- Avoid service within a service
Preparing GIS Documents

*Dynamic Map Services: Text and Labeling*

- Avoid Maplex for dynamic labeling
- Use annotation instead of labels when possible
- Avoid special effects (fill patterns, halos, callouts, backgrounds)
- Avoid very large text size (60+ pts)
- Avoid Highway symbols
- Use indexed fields (reduce label SQL query number and complexity when possible)
- Use label and feature conflict weights sparingly
- Avoid overuse
## Performance Factors: ArcGIS Server Services

**Demo: Optimizing `mxd` with `MXDPerfStat` (ArcScripts)**

<table>
<thead>
<tr>
<th>Item</th>
<th>At Scale</th>
<th>Layer Name</th>
<th>Refresh Time (sec)</th>
<th>Recommendations</th>
<th>Features</th>
<th>Vertices</th>
<th>Labeling</th>
<th>Geography Phase (sec)</th>
<th>Graphics Phase (sec)</th>
<th>Cursor Phase (sec)</th>
<th>DBMS CPU</th>
<th>DBMS LIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>8,000</td>
<td>Tax Lots</td>
<td>1.05</td>
<td>Simplify labeling, symbology: <code>GraphicsPhase=.83</code>;</td>
<td>2,226</td>
<td>33,872</td>
<td>True</td>
<td>.14</td>
<td>.83</td>
<td>.20</td>
<td>.08</td>
<td>6,396</td>
</tr>
<tr>
<td>19</td>
<td>8,000</td>
<td>Tax Lots Query Def</td>
<td>.13</td>
<td></td>
<td>1</td>
<td>26</td>
<td>False</td>
<td>.03</td>
<td>.02</td>
<td>.06</td>
<td>.03</td>
<td>3,204</td>
</tr>
<tr>
<td>20</td>
<td>8,000</td>
<td>TaxlotDenseLabel</td>
<td>1.84</td>
<td>Simplify labeling, symbology: <code>GraphicsPhase=1.03</code>; simplify geometry and/or set label scale; convert polygon to polyline: vertices fetched=200001; simplify geometry and/or set label scale: vertices fetched=200001;</td>
<td>1</td>
<td>200,001</td>
<td>True</td>
<td>.73</td>
<td>1.03</td>
<td>.95</td>
<td>.01</td>
<td>266</td>
</tr>
<tr>
<td>21</td>
<td>8,000</td>
<td>TaxlotDenseNoLabel</td>
<td>.53</td>
<td>simplify geometry: vertices fetched=200001;</td>
<td>1</td>
<td>200,001</td>
<td>False</td>
<td>.47</td>
<td>.02</td>
<td>.97</td>
<td>.00</td>
<td>140</td>
</tr>
</tbody>
</table>
Optimized Map Services

New at 9.3.1

• New high performance drawing engine
  – Completely re-designed
  – Focused on symbology and cartography

• Provides best performance for dynamic maps

• Build cache faster

• Creates a service definition file (MSD)
Map Service Publishing Toolbar

*New at 9.3.1*

- Identify layers or issues that may degrade performance
- Available in ArcView, ArcEditor and ArcInfo
- Analyze and preview in ArcMap
Analyze the map

Improve performance

• Discover content affecting performance

• Review results: Errors, Warning and Messages

• Preset solutions available in the analysis tool
Preview the map

Improve performance

- Assess drawing performance and graphic quality before publishing
Performance Factors: ArcGIS Server Services (MapService)

Uncorrected vs. Corrected Source ArcMap Document

Throughput (Transactions/Hr)

Corrected
(Unloaded TT: 0.34 sec)
(2.1 Instances/core)

Uncorrected
(Unloaded TT: 11.97 sec)
(1.6 Instances/core)
Preparing GIS Documents

Map Services

• Caching always gives best performance

• Use optimized map services for operational layers

• Use standard map services for layers that cannot be optimized
Preparing GIS Documents

Geoprocessing Services

• Use Local Jobs Directory
  – Greatest single performance factor.
  – 9.3.1 allows simple deployment

• Subset very large rasters

• Use native types like GRID

• Use Layers instead of raw data from db

• Use in-memory feature data sets
Preparing GIS Documents

Geoprocessing Services

- Pre-compute intermediate steps when possible
- Use local paths to data and resources
- Avoid unneeded coordinate transformations
- Add attribute indexes
- Simplify data

Desktop Help:

Performance tips for geoprocessing services
Preparing GIS Documents

Mobile Services

- **Document Preparation**
  - Minimize operational layers
  - Cache basemap layers

- **Service Configuration**
  - Try to keep total service cache size under 250 MB to avoid network swapping.
  - Upload/Download size limits (200K upload/4MB download)

- **Usage considerations**
  - Avoid batch postings in favor of frequent updates
Preparing GIS Documents

Geocoding Services

- Single address geocoding
  - ArcSDE address locators for single address geocoding

- Batch address geocoding
  - File-based address locators for batch geocoding
    - Use local locator files instead of UNC

- All geocoding
  - Take locator defaults
Preparing GIS Documents

Geodata Services

• Data Preparation
  – Perform regular version maintenance (keep versioning tree small, compress, schedule synchronizations, etc)
  
  – Use 1 way replicas over 2 way replicas when possible

  – Consider 2 way replicas instead of check-out replicas

  – Well-defined data model
Topical Outline

• Preparing GIS documents
• Publishing GIS services
• Creating and Configuring GIS applications
• Administration & System Monitoring
• Deployment Architectures
Publishing GIS Services

- Response Time Factors
- Object Pooling and Instances
- Process Isolation
- Mapping
  - Caching
  - Output Image Types
- Mobile Caches
- Geoprocessing
- Geocoding
Publishing GIS Services

Response Time Factors

- **Response Time = Service Time + Queue Time**

  - **Service Time** represents CPU processing at the various tiers:
    - Browser Client CPU Processing
    - Web Server CPU Processing
    - Application Server CPU Processing
    - Data Server CPU Processing

- **Queue Time** represents time delay:
  - Network Latency
  - Disk Access
  - Application Queue
  - CPU Queue

---

**Typical Response Time Factors**
Publishing GIS Services

**Object Pooling**

**Pooled Services**
- State information (e.g., Current extent, layer visibility, etc.) maintained in web server / browser
- **Scales better**

**Non-Pooled Services**
- Typically holds its reference to the service for the duration of the application's session
- Number of users on the system can have no more than a 1:1 correlation with the number of running service instances
- Required for Editing Task ADF
- Computationally expensive – New instances are started for each session
Publishing GIS Services

Object Pooling

- Define Min-Max instances
- Instances are distributed across all host servers
Publishing GIS Services (Mxd)

Object Instances

**Recommendation:** Limit the number of service instances to 2-4 instances per CPU core (typical).
Publishing GIS Services

*Process Isolation*

- **High Isolation:** each service instance runs as a single thread in a dedicated ArcSOC.exe process.

- **Low Isolation:** many service instances run as separate threads in each ArcSOC.exe process.

- **Recommendation:** Use high isolation
  - A failed instance is “isolated” to one ArcSOC.exe process

**Note:** At 9.2, low isolation max threads is 4. At 9.3, default is 8 and adjustable.
Publishing GIS Services

Cache Creation

- Can be a very time consuming
  - Build it for a small area to test
  - Use “Cache On Demand”
  - Cache configuration available through ArcGIS Server Manager
- For cache tile sizing – use default 512 x 512. Smaller tiles sizes increase end-user network traffic & consume more disk space.
- For higher aesthetics, use Anti-Aliasing.

<table>
<thead>
<tr>
<th>Tile Size</th>
<th>Files</th>
<th>Size on Disk</th>
<th>Creation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>512x512</td>
<td>19,000</td>
<td>0.2 GB</td>
<td>1 hour</td>
</tr>
<tr>
<td>128x128</td>
<td>311,000</td>
<td>1.2 GB</td>
<td>5 hours</td>
</tr>
</tbody>
</table>
Publishing GIS Services

Cache Creation

- Anti-aliased tiles are rendered at finer resolution followed by down sampling
  - Smoothes the edges of labels and lines by blending them with the background.
  - The resulting screen display quality is better than standard rendering in ArcMap.
  - Cache generations times can double with Anti-aliasing enabled.
Cache Creation

- Cache generation runs as a “batch process” and for the most part consumes a CPU.
- Recommended number of SOC processes is N+1, where N is the number of available CPU cores.
- The additional SOC will consume any “left-over” CPU cycles while busy SOCs are waiting on data.
- Too many SOC processes will cause CPU competition.
Publishing GIS Services

Output Image Types

- Output image size varies by format and underlying data type. Examples of 600 x 400 pixel images

<table>
<thead>
<tr>
<th>Format</th>
<th>Size (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG</td>
<td>76</td>
</tr>
<tr>
<td>PNG24</td>
<td>316</td>
</tr>
<tr>
<td>BMP</td>
<td>703</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Format</th>
<th>Size (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG</td>
<td>70</td>
</tr>
<tr>
<td>PNG24</td>
<td>30</td>
</tr>
<tr>
<td>BMP</td>
<td>703</td>
</tr>
</tbody>
</table>

• Recommendation:
  - Use appropriate output type to support applications while minimizing impact to network. Generally, raster data is best served in a JPEG format, while vector data is best served in a PNG format. Use PNG32 to support transparency.
Publishing GIS Services

*Mobile Cache*

- Build map cache slightly larger than the extent your field worker will be working at.
- Build cache for the entire extent of your background layers (avoids synchronization)
- Keep background data as small as possible (device storage issues)
- Load cache onto devices *before* going out into the field (provisioning).
Publishing GIS Services

Geoprocessing

- If possible use **Execution Type = Synchronous**
Publishing GIS Services

Batch Geocoding

- Default batch size is 10. Change to between 100 and 400.

- At 9.3 default batch size is 1000 and should be set no more than 2000.
Topical Outline

• Preparing GIS documents
• Publishing GIS services
• Creating and Configuring GIS applications
• Administration & System Monitoring
• Deployment Architectures
Creating & Configuring GIS Applications

Connecting to ArcGIS Server

- Clients Applications should use **Internet Connections** to ArcGIS Server unless Local Connections are required (e.g. Web Editing Task).

![ArcGIS Desktop Dialog](image1)

![ArcGIS Server Manager Dialog](image2)
Creating & Configuring GIS Applications

Web ADF Map Viewer Application

• 9.2 SP3
  – .NET memory utilization has been improved.
  – .NET TOC issues have been resolved.

• 9.2 SP4
  – Overview Map no longer causes significant performance impact

For Web ADF, custom application that does not require seamless pans, set the Properties of the Map control:
  – EnableTileCaching = FALSE
  – EnableContinuousCallback = FALSE
Creating & Configuring GIS Applications - ADF

**Blending of Cache and Dynamic Services**

- Performance for browser blending in .NET is slow for 9.2 (Java does not have this issue).
- For 9.2, server-side blending in the SOC is more stable and supports more concurrent users.
- For 9.3, browser-side blending is the recommended approach and much higher performing.
Creating & Configuring GIS Applications/Services

LSASS CPU & Memory

- Local Security Authentication Server system process (lsass.exe) grows in CPU usage and memory utilization under heavy load.
  - Solution:
    - Create a new IIS Application Pool and set its identity to the ArcGIS Web Services account.
    - Set the ArcGIS Web Services and REST applications to not use impersonation.
    - Add the ArcGIS Web Services and REST applications to the newly created application pool.
    - Windows_Server_2003

- [http://support.esri.com/index.cfm?fa=knowledgebase.techarticles.articleShow&d=32622](http://support.esri.com/index.cfm?fa=knowledgebase.techarticles.articleShow&d=32622)
Creating & Configuring GIS Applications/Services

MapService

Map URL: http://andrew08/arcgis/rest/services/Taxlot/MapServer

Query Task URL: http://andrew08/arcgis/rest/services/Taxlot/MapServer/0

Status

1914 ms: Application creation completed.
63834 ms: Starting query...
65502 ms: Query took 1667 ms [999 features].
66021 ms: Graphics drawn in 519 ms [999 features].
Tiles took 1883 ms to zoom.
Tiles took 2844 ms to pan.
Topical Outline

• Preparing GIS documents

• Publishing GIS services

• Creating and Configuring GIS applications

• Administration & System Monitoring

• Deployment Architectures
Performance Tips

**Database**

- Optimize and tune your database
- Use Direct Connect to connect your map service to your database
- Store file-based data (i.e., file geodatabase) on the SOC server
- Use Spatial and Attribute indexes
ArcGIS Server Administration

Administration Tools

- **ArcCatalog**
  - Administer ArcGIS Server
  - Consume ArcGIS Server services

- **Server Manager**
  - Web Based Interface
  - Publishes Services
  - Administer ArcGIS Server
  - Creates Template-Style Web Applications
  - Troubleshoot Using Logs
  - Secure Web applications and services (9.3)
ArcGIS Server Monitoring

Service Usage Time Statistics (ArcCatalog)

- **Usage Time**
  - Start to finish time for service instance use (service processing time)
  - Mostly useful for pooled services
ArcGIS Server Monitoring
Service Creation Time Statistics (ArcCatalog)

• **Creation Time**
  - Amount of time to start the service (load the MXD, etc.)
ArcGIS Server Monitoring

Service Wait Time Statistics (ArcCatalog)

- **Wait Time**
  - Amount of time a client waits to get access to a service instance (SOM queue time)
**ArcGIS Server Log Files**

**Manager Logs**

- Use “Normal” mode for standard operations
- Use “Detailed” or “Debug” mode for troubleshooting

**Default Log Location:** C:\Program Files\ArcGIS\server\user\log
System Monitoring

Operating System Tools

- Windows – Perfmon

- Linux – DSTAT, IOSTAT, VMSTAT, PS

- Unix – DSTAT, NMON, IOSTAT, VMSTAT, PS
System Monitoring
HTTP Conversation

- Microsoft’s Fiddler
  - HTTP debugging proxy
  - http://www.fiddlertool.com/
Topical Outline

- Preparing GIS documents
- Publishing GIS services
- Creating and Configuring GIS applications
- Administration & System Monitoring
- Deployment Architectures
Deployment Architectures

Adding Capacity

Single Machine Scenario

Multiple Machine Scenario
Deployment Architectures

Small Capacity Enterprise (Single Machine)

Clients

Reverse Proxy

ArcGIS Server

Comments:

- Simple Setup

- Use Application Server (AS) connections if RDBMS is dedicated.

- Use local drives for FGDB/SHP data and output directories
Deployment Architectures

**Large Capacity Enterprise (Multiple Machine – Simple)**

- Network Load Balancer

**Web Tier**
- SOM
- SOC

- **Advantages:**
  - Few bottlenecks
  - Easy scaling

- **Disadvantages:**
  - Better throughput possible (SOCs burdened by other components)
  - Must replicate configurations
  - Must replicate Security
  - Sticky sessions if state required
Deployment Architectures

Large Capacity Enterprise (Multiple Machine - MIME)

Advantages:
- Highest Throughput
- Single service config

Disadvantages:
- Must replicate security
- Network Bottlenecks

WebServices/ADF (WS1)

Network Load Balancer

SO (1)  SO (2)  ...  SO (N)

ArcSDE DC
Deployment Architectures

Adding Additional Servers

ArcGIS Server Properties

The list below shows the machines available to host services.
NOTE: you need to add at least one machine to use the server.

<table>
<thead>
<tr>
<th>Machine Name</th>
<th>Description</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>napoleon14</td>
<td></td>
<td>&lt;Unlimited&gt;</td>
</tr>
</tbody>
</table>

Add Machine

- Machine Name: New_Machine
- Capacity (Maximum number of instances): <Unlimited>
- Description:
Deployment Architectures

Firewalls

- ESRI does not recommend firewalls between ArcGIS Server components but rather recommends the use of a Reverse Proxy web server for securing access to Trusted systems. 

For example:

```
Intranet

Trusted systems
ArcGIS Server
(Web Tier, SOM, SOC, ArcSDE)

DMZ

Web servers
Reverse Proxy
Web Server

Internet

Non-Trusted systems
```

More Information at:
- Server Help: Firewalls and Server
- Support KB Article
Deployment Architectures

*Capacity Planning*

- Use your testing and staging landscapes (your data, applications and hardware)

- ESRI Professional Services Enterprise Consulting


Conclusions

- ArcGIS Server offers a powerful and flexible platform for server-based GIS, but it needs to be architected correctly.

- Preparing GIS documents
- Publishing GIS services
- Creating and Configuring GIS applications
- Defining a deployment architecture
More Information

SERUG
- Designing, Deploying, and Using Cached Map Services  Wed 8:30am

Seminar
- Creating Effective Web Maps
  - Columbia, SC 4/30/2009
  - Richmond, VA 5/7/2009
  - Nashville, TN 5/19/2009
  - Orlando, FL 5/21/2009
  - Atlanta, GA 6/4/2009
  - Birmingham, AL 6/9/2009
More Information: Classroom

• Instructor-Led Training Courses

- Introduction to ArcGIS Server

- ArcGIS Server: Web Administration Using the Microsoft .NET Framework

- Coming soon: Building Web Maps Using the ArcGIS API for JavaScript
More Information: Online

Resource Center: DS2009

– Best Practices for Designing Effective Map Services

– Advanced Map Caching Topics

– ArcGIS Server Performance and Scalability—Performance Factors and Optimization

– ArcGIS Server Performance and Scalability—Testing Methodologies
Thank You
and
Enjoy the Conference!