Architecting ArcGIS Server Solutions for Performance and Scalability

Dave Wrazien

Eric Miller
Schedule

• 75 minute session
  – 60 – 65 minute lecture
  – 10 – 15 minutes Q & A following the lecture

• Cell phones and pagers

  Please!
  Turn OFF cell phones and paging devices

• Please complete the session survey – we take your feedback very seriously!
Introductions

• Who are we?
  – Dave Wrazien – Solution Architect
  – Eric Miller - ArcGIS Server Framework Product Engineer

• Who are you?
  – New to ArcGIS Server?
  – Currently deploying ArcGIS Server 9.2?
  – Currently deploying ArcIMS?
Topical Outline

- Preparing GIS documents
- Publishing GIS services
- Creating and Configuring GIS applications
- Deployment Architectures
Preparing GIS Documents

General Overview

• Understand application requirements
• Design specifically for server deployment
  – Maps
    • Dynamic – “operational” data
    • Cache – “basemap” data
  – Geoprocessing Models
    • Preprocessing
    • Limit inputs
  – Mobile Applications
    • Devices
    • Environment
  – Geocoding
    • Locator locations
Preparing GIS Documents

*Dynamic Map Services: General Information*

- **Show relevant information**
  - Start simple (additional layers can be toggled on by user)
  - Use field visibility (hide unnecessary attributes)
- **Use scale dependencies**
  - Use data appropriate for the given scale (generalize if necessary)
  - Display similar number of features at all scales for consistent user experience
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, complex shapes, masking

• Line & Polygons
  – Use ESRI_Optimized style
  – Avoid cartographic lines (also includes polygon outline!)

Preparing GIS Documents

*Dynamic Map Services: Text and Labeling*

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

Geoprocessing Services

- Understand Performance Expectations
- Simplify Models and data
  - Preprocess steps in advance.
- Use in-memory data
- Use fast-access data (uncompressed).
- Two instances cannot update the same data at the same time.
Preparing GIS Documents

**Mobile Services**

- **Design for Purpose**
  - Understand workflow and tasks
  - Understand user skills & familiarity
  - Deliver only relevant content

- **Design for Environment**
  - Changes in lighting impact colors
  - Contrast (use gray scales)
  - Simplicity (avoid polygon fills)

- **Design for Form Factor**
  - Resolution (Smartphone = 320x240 & Tablet PC =1024x768+)
  - Storage (keep background data small)
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
Topical Outline

• Preparing GIS documents

• Publishing GIS services

• Creating and Configuring GIS applications

• Deployment Architectures
Publishing GIS Services

- Object Pooling and Instances
- Process Isolation
- Mapping
  - Caching
  - Output Image Types
- Mobile Caches
- Geoprocessing
- Geocoding
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
- 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

Object Instances

Recommendation: Limit the number of service instances to 2-4 instances per CPU (typical).

Too many instances creates CPU competition

Too few instances creates resource deficiency
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
Publishing GIS Services

**Caching**

- Pre-rendering and storing of map images (tiles) for rapid, high performance display.
- Clients can still access the underlying data of the cached map service
  - Identify, Query, Highlight feature geometry

**Recommendation:** Use Cache map services whenever possible... especially for basemaps.
Publishing GIS Services

Cache Creation

• Can be a very time consuming
  – Build it for a small area to test
  – Use “Cache On Demand”

• 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>
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.
Publishing GIS Services

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

  **Raster and Vector Data**
  - JPEG = 76 KB
  - PNG24 = 316 KB
  - BMP = 703 KB

  **Vector Only Data**
  - JPEG = 70 KB
  - PNG24 = 30 KB
  - BMP = 703 KB

**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.
Publishing GIS Services

Geoprocessing

- If using ArcSDE data sources, be sure to install SP4.
- If possible use **Execution Type = Synchronous**
Publishing GIS Services

**Geocoding**

- Default batch size is 10. Change to 100.

- If batch size is more than 400 serialization/deserialization may have negative performance impact

- At 9.3 default batch size is 1000 and serialization/deserialization limit will be set to 2000.
Topical Outline

• Preparing GIS documents

• Publishing GIS services

• Creating and Configuring GIS applications

• 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

ArcGIS Server Manager Dialog
Creating & Configuring GIS Applications

Web Map Viewer Application

• Between SP2 and SP3, .NET memory utilization has been improved. 9.2

• Between SP2 and SP3, .NET TOC issues have been resolved. 9.2

• The Overview Map provides the most significant performance impact. (omit if possible or use static Overview Map in SP4) 9.2

• If building a custom application that does not require seamless pans, set the EnableTileCaching = FALSE and EnableContinuousCallback = FALSE (default values are TRUE) properties of the Map Control. 33% increase in overall throughput can be achieved.
Creating & Configuring GIS Applications

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

9.2 Server-side Blending

Authored MXD
Creating & Configuring GIS Applications

Delivery to Applications

• Consider HTTP Compression for faster file download….

  (IIS)
  – UseDefaultWebResources = False
  – WebResourceLocation = <virtual directory>

• MIME Data… by default UseMimeData = TRUE, can set to FALSE if a public virtual directory is permissible.
Topical Outline

- Preparing GIS documents
- Publishing GIS services
- Creating and Configuring GIS applications
- Deployment Architectures
Deployment Architectures

Adding Capacity

Single Machine Scenario

Multiple Machine Scenario
Deployment Architectures

Adding Additional Servers

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>
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.

Deployment Architectures

Recommended Deployments

- ArcSDE
- SOC
- SOM
- Workgroup
- DBMS
- Web Tier

Workgroup
Deployment Architectures

Recommended Deployments

• **Configuration**
  - ArcGIS Server Workgroup
  - Reverse Proxy
Deployment Architectures
Recommended Deployments

- **Configuration**
  - ArcGIS Server Workgroup
  - Reverse Proxy
Deployment Architectures

Recommended Deployments: ArcGIS Server Workgroup

- **Advantages**
  - Fully functional

- **Disadvantages**
  - Reverse Proxy must be configured with filters for optimal security.
  - Internal users must be able to resolve the external URLs otherwise two sets of services are required (one internal and one external).
  - Single points of failure exist.
Deployment Architectures
Windows Workgroup Configuration

• If installing in a Windows Workgroup (not a Domain)
  – Simple File Sharing must be “Off” on XP.
  – Core ArcGIS Server accounts must be local users.
  – Local Security Policy Setting
    • For “Network access: Sharing and security model for local accounts” Set to “Classic – local users authenticate as themselves”.
Deployment Architectures

Recommended Deployments

Web Tier
SOM
SOC
ArcSDE

Direct Connect

DBMS

Small Capacity Enterprise
or Standard Development/Testing Environment
Deployment Architectures

*Recommended Deployments*

- **Configuration**
  - Small Capacity Enterprise
  - Reverse Proxy
Deployment Architectures

Recommended Deployments

• Configuration
  – Small Capacity Enterprise
  – Reverse Proxy
Deployment Architectures

Recommended Deployments: Small Capacity Enterprise

• **Advantages**
  – Fully functional
  – Improved performance by moving RDBMS to additional machine.

• **Disadvantages**
  – Reverse Proxy must be configured with filters for optimal security.
  – Internal users must be able to resolve the external URLs otherwise two sets of services are required (one internal and one external).
  – Single points of failure exist.
Deployment Architectures

Recommended Deployments

Small Capacity Enterprise, High Availability
or Standard Production Environment
Deployment Architectures

Recommended Deployments

- **Configuration**
  - Small Capacity Enterprise
  - High Availability
  - Network Load Balancer
  - Reverse Proxy

- **Configuration**
  - Small Capacity Enterprise
  - High Availability
  - Network Load Balancer
  - Reverse Proxy
Deployment Architectures

Recommended Deployments

- Configuration
  - Small Capacity Enterprise
  - High Availability
  - Network Load Balancer
  - Reverse Proxy

Identical communications on each machine.
(not shown on additional machines for clarity)
Deployment Architectures

Recommended Deployments: Small Capacity Enterprise, High Availability

• **Advantages:**
  – Fully Functional
  – Failover safe
  – Full SOC resource utilization (each SOM uses both machines as Service Hosts) even after failure of a SOM or WebTier component.
  – No single points of failure

• **Disadvantages:**
  – Web Tier gets heavily loaded after failover.
  – Reverse Proxy configuration must be duplicated on each DMZ web server.
  – Reverse Proxy must be configured with filters for optimal security.
  – Internal users must be able to resolve the external URLs otherwise two sets of services are required (one internal and one external).
  – Services must be duplicated for each SOM.
  – ADF web applications must be duplicated on each Web Tier.
Deployment Architectures

Recommended Deployments

Medium Capacity Enterprise, High Availability
Deployment Architectures

**Recommended Deployments**

- **Configuration**
  - Medium Capacity Enterprise
  - High Availability
  - Web Tier, SOM, RDBMS and Reverse Proxy Failover
Deployment Architectures

Recommended Deployments: Medium Capacity Enterprise, High Availability

• **Advantages:**
  – Fully Functional
  – Failover safe
  – Full SOC resource utilization (each SOM uses both machines as Service Hosts) even after failure of a SOM or WebTier component.
  – No single points of failure

• **Disadvantages:**
  – Web Tier gets heavily loaded after failover.
  – Reverse Proxy configuration must be duplicated on each DMZ web server.
  – Reverse Proxy must be configured with filters for optimal security.
  – Internal users must be able to resolve the external URLs otherwise two sets of services are required (one internal and one external).
  – Services must be duplicated for each SOM.
  – ADF web applications must be duplicated on each Web Tier.
Deployment Architectures

Recommended Deployments

Large Capacity Enterprise, High Availability
Deployment Architectures

Recommended Deployments

• Configuration
  – Large Capacity Enterprise
  – High Availability
  – Web Tier, SOM, RDBMS and Reverse Proxy Failover

- ArcGIS Server Machine
  - Web Tier
  - ADF Apps
  - Manager
  - Services
  - Application Server Tier
  - SOM

- ArcGIS Server Machine
  - Web Tier
  - ADF Apps
  - Manager
  - Services
  - Application Server Tier
  - SOM

- ArcGIS Server Machine
  - Application Server Tier
  - SOC
  - ArcSDE DC

- ArcGIS Server Machine
  - Application Server Tier
  - SOC
  - ArcSDE DC

- ArcGIS Server Machine
  - Application Server Tier
  - SOC
  - ArcSDE DC

- ArcGIS Server Machine
  - Application Server Tier
  - SOC
  - ArcSDE DC

- DBMS (Configured for Failover)

- Firewall

- Network Load Balancer (Configured for failover)

- Web Server (Reverse Proxy)

- Web Server (Reverse Proxy)

DMZ

Internal Network
Deployment Architectures

Recommended Deployments: Large Capacity Enterprise, High Availability

- **Advantages:**
  - Fully Functional
  - Failover safe
  - Full SOC resource utilization (each SOM uses both machines as Service Hosts) even after failure of a SOM or WebTier component.
  - No single points of failure

- **Disadvantages:**
  - Web Tier gets heavily loaded after failover.
  - Reverse Proxy configuration must be duplicated on each DMZ web server.
  - Reverse Proxy must be configured with filters for optimal security.
  - Internal users must be able to resolve the external URLs otherwise two sets of services are required (one internal and one external).
  - Services must be duplicated for each SOM.
  - ADF web applications must be duplicated on each Web Tier.
Goal: How many users can my service support on a single SOC machine?

Procedure:

- Create a system load model
  - 500 users
  - 10% peak concurrency
  - 50 concurrent users (10% of 500)
  - User workflow of 6 GIS transactions (displays) per minute – implies transaction time + client think time = 10 seconds
    - Assuming clients need 5 seconds to process an image, client think time is 5 seconds
    - Implies transaction time cannot be more than 5 seconds

- Determine maximum throughput of a single SOC machine for your service.

- Determine maximum number of concurrent clients supported on a single SOC machine with a maximum acceptable transaction time.

- Multiply this out to the number of concurrent clients you need to support.
Deployment Architectures

*Determining Throughput*

**Goal:**

Determine the throughput of a representative service at the point where the system becomes fully loaded.
Deployment Architectures

Determining Throughput

Procedure:

• Fix Minimum and Maximum instances to an initial guess of 3-4 instances per CPU core.

• Load the service with constant load clients.
  – Start with one client loading one service instance and collect 50-100 transactions (extrapolate to transactions/hour).
  – Repeat until clients equal the number of service instances.
  – Produce curve of Throughput vs loaded service instances.
  – Pick point after throughput plateaus as the maximum number of instances the service can support.

• Refine Minimum/Maximum instances if machines are over or under loaded.
Deployment Architectures

*Determining Number of Clients Supported with Acceptable TT*

**Goal:**

Determine the maximum number of clients a single SOC machine can support with a maximum acceptable transaction time.

![Graph showing theoretical average transaction time per client vs. number of clients, with a line indicating maximum acceptable transaction time and a point at approximately 15 clients.](image-url)
Deployment Architectures

Determining Number of Clients Supported with Acceptable TT

Procedure:

• Set the min/max instances of your service to the number determined in the Throughput Test.

• Increase client load (with think time) one instance at a time until the average response time exceeds what the user can tolerate. This is the number of real-world clients your machine can handle.

• Scale your system by adding machines until you have enough to meet expected peak concurrent user load.
Deployment Architectures

Capacity Planning

- 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

• Other recommended sessions and meetings:
  – “Implementing and Optimizing ArcGIS Server Map Caches”
  – “Building and Optimizing Geoprocessing Services for ArcGIS Server”
  – “Implementing Security for ArcGIS Server .NET Solutions”
  – “Implementing Security for ArcGIS Server Java Solutions”
  – “Architecting ArcGIS Server Solutions for Linux and Solaris”

• All sessions are recorded and will be available on EDN

• Still have questions?
  1. Tech talk, Demo Theatres, Meet the Team
  2. “Ask a Developer” link on web page
     • http://www.esri.com/devsummit/techquestions

• Please fill out session surveys!
Thank You and Enjoy the Conference!