WELCOME
2016 Southeast User Conference
ArcGIS for Server Performance and Scalability

David Crosby
Esri Professional Services
dcrosby@esri.com
Workshop Overview

- Architecture
- Map and Image Services
- Tuning ArcGIS Server
  - System Monitor and server statistics
- What’s new

Goal: Provide valuable information that you may take back to your organization and improve your implementation of ArcGIS Server

** You do not want to see performance problems in production! **
Architecture
Simplest - Sandbox
Introduction of third-party load balancers
Additional Considerations – Publication Geodatabases
Additional Considerations – Workload Separation

Figure 1: GIS Server machines organized within sites which are separated by business function.
Additional Considerations – Customary Environments

Figure 1: Three recommended computing environments.
Increasing Capacity
Difference between increasing performance and increasing capacity

• Multi-machine GIS Server site could increase performance, will definitely increase capacity and high-availability

• Performance will possibly increased, depending on what was causing the issue in the first place

• If a server is improperly tuned and does not have enough instances, memory and tuning can help, in addition to adding an additional server to the site

• How do we know? Monitoring – looking at wait timeouts
Optimizing map services
Overview

- Types of map services
  - What's new in the last few releases
- Factors of map service performance
  - Data access
  - Network
  - Server
  - Client Rendering
  - Image size

  **Consider everything that it takes to return a result from a service as additive and minimize all of the pieces**
Map services – a brief history

• 9.3.1 – 10.0
  - MXD based map services
  - MSD based map services (optimized map service)
  - MSD’s brought performance of the GIS server up to that of ArcIMS

• Recommendations through version 10 were to use the optimized map service
Map services at 10.1 and beyond

• One unified map service
  - An updated optimized map service
  - Supports additional capabilities, data types, layers, renderers

• Includes extension capabilities optimized map service lacked:
  - Network Analysis
  - Geoprocessing*
Mapping capabilities

- Added data source support
  - XY events
  - Linear referencing events

- Added feature layer renderer support
  - Dot density
  - Charts
  - Geostats

- Added support for layers
  - Dimensions
  - Schematics
  - Network
  - Network Analysis
  - Tin
  - Terrain
  - Tracking
Dynamic Layers

- New behavior with the map service that allows for per-request changes to the map
  - Optional capability of map services
- May allow you to reduce the total number of services you need
- Fewer services = higher performing site

- Allows for:
  - Updating renderers and symbols
  - Removing and reordering layers
  - Changing layer data sources
  - Adding new layers from registered data sources
Dynamic Layers: How they work

- Simple stateless requests to the service through REST

- Simple updates to the map service
  - Remove layers or reorder layers

- Thematic mapping
  - Updates to renderers

- Adding content from your datasources
  - Find data from registered workspaces
    - Including query layers
  - Add to the map on a per-request basis
Factors of map service performance

- Data access – often most important
- Content of map
- Rendering performance
- Image compression / size
  - PNG 32 large, PNG 8 smaller

- Consider all of these when creating a map service
- Every step is cumulative in nature and will contribute to total response time
Cached Map Service Data Flow

Client App

Client request tile

Server

(Search/Loadbalancer/TileHandler)

Search Tile Index

Read tiles from storage

≈20KB

Return Tile

Bundle Tile Cache

Décompressive tile & display

Cache Tile Locally

Bottleneck? : Typically Network

≈20KB
Dynamic Map Service Data Flow

Client App

Client request (image, geometry)

Typically Network

Typically Network

Bottleneck? : Network

Send query to data/geodatabase

Read data from storage

Return Image (MS) or Geometry (FS)

≈20KB

Return Data

Display image

Browser renders features

Server (Security/Loadbalancer/)

Storage

Geodatabase/RDBMS
Data access

- Local data will draw faster than remote data
- UNC path vs. local drive
- Spatial index
  - Do you have one? (e.g. XY Events)
  - Is it sized correctly?
    - Updated after data load?
- Attribute indexing
  - Not always needed
Data access case study: XY Events

- Often used in cases where data comes from external systems
  - As database table or CSV
  - They are easy, so used commonly – but not indexed
- A draw typically requires a complete row scan
- One of the slowest ways to draw the data

- Alternative
  - Use a native spatial type in your database
    - Query layers
  - Insert features via SQL in external systems
  - Features will be indexed and draw much faster
Map Service Troubleshooting

• Publishing analyzers indicate lack of a spatial index etc.

• Pay attention to and address all of the suggestions made in the analyzer

• Evaluate I/O performance if using remote data
  - How much, how fast is the network?

• Unnecessary attribute indexes
  - May confuse query plans in some databases
  - Don’t index fields just because they exist in a definition query
Rendering speed

- Remaining areas to be concerned with:
  - Complex effects (e.g. geometric effects in representations)
  - Maplex labels vs. standard labeling – use annotation where you can
  - Anti-aliasing performance
    - Higher levels use more RAM and are slower
    - Text anti-aliasing has a negligible effect in most cases
    - Maps DO look much better, must make a decision. Consider caching.
  - Layer transparency
Rendering speed: transparency

- Layer transparency is applied to a layer as a whole
  - Involves a full layer blend
  - Draws ENTIRE layer, then blends with existing content

- Alternative: Use color transparency
  - Capability of optimized map services – read the help, will not work for overlapping features
  - Enabled via an option from analyzer warning 10009
Image compression / size

- Smaller images are faster to download

- Image compression itself has a performance penalty
  - Use the preview window to evaluate performance of image type
  - Try out different image types in there – look at type and time to draw
  - Try BMP – never use in production, but compare it to other types
Demo

- Analyzer and preview window
Image compression / size (con’t)

- **Image type used for cached services affects:**
  - Download size
  - Storage size of the cache
    - Portability

- **For caching vector data**
  - Consider new PNG image type (introduced at 10.1)
  - Chooses the correct PNG type (8, 24, 32) for each tile based on content
  - Low content areas use less storage
Recommendations:

- General
  - If just want static background Use Tile Cache
  - Do not add imagery to a Map and Publish as Map. (Best practice is to keep image as separate service)
  - Cache wherever possible and script updates to areas of interest
  - Feature Access if editing is required, and only if it is required
Storage Infrastructure

Ensure fast transfer to Servers

• Disk Storage – usually the biggest bottleneck
  - Needs to be Fast
  - Stripe Disks (more data accessed simultaneously)
  - Tune NAS. Check turning off Read Ahead Cache?
  - Storage performance can vary significantly

• Internal Network
  - Min 1GB between server and storage
  - If necessary use dedicated network for imagery
Server

- CPU – Faster Better
- Memory – 3 GB/Core (Linear relationship between ArcSOC.exe and memory)
- Virus Checker – Can be a real burden!
## Optimize Services

### Pooling tab

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>When to adjust it</th>
</tr>
</thead>
</table>
| Minimum number of instances per machine      | Minimum number of instances of the service to create on each node within the cluster. Default = 1 | • Increase if there are frequently more requests than available instances.  
  • Decrease if there are frequently more available instances than requests. |
| Maximum number of instances per machine      | Maximum number of instances of the service to create on each node within the cluster. Default = 2 | • Increase if clients are frequently waiting for an available instance.  
  • Decrease for infrequent requests. |
| The maximum time a client can use a service  | Maximum number of seconds an instance can service a request. Default = 600 seconds (10 minutes) | • Increase for requests that may take longer (e.g., geoprocessing jobs).  
  • Decrease for services that complete quickly to free system resources. |
| The maximum time a client will wait to get a service | Maximum number of seconds the framework will wait for a free instance of the service before rejecting a service request. Default = 60 seconds | • Increase on a busy system. |
| The maximum time an idle instance can be kept running | Maximum number of seconds an idle instance of a service must be kept alive before it can be destroyed. Default = 1800 seconds (30 minutes) | • Increase if new instances are created frequently.  
  • Decrease for infrequently requested services to free system resources. |
## Optimize Services

### Processes tab

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>When to adjust it</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recycling Settings</td>
<td>Duration between service recycling.</td>
<td>• Decrease to reclaim resources and clear HTTP connection issues.</td>
</tr>
<tr>
<td></td>
<td><strong>Default = 24 hours at 00:00</strong></td>
<td>• You require that a service recycles at a different time of day.</td>
</tr>
<tr>
<td>Health Checks</td>
<td>Duration between data connections check &amp; repair (idle instances).</td>
<td>• Decrease if SDE connections frequently break.</td>
</tr>
<tr>
<td></td>
<td><strong>Default = 30 minutes</strong></td>
<td></td>
</tr>
</tbody>
</table>
## Optimize Services

### Parameters tab

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>When to adjust it</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Number of Records Returned by Server</td>
<td>Maximum number of records returned for a service request. <strong>Default = 1000</strong></td>
<td>- Increase if the service will be returning large numbers of records when fulfilling requests. This will increase network usage and possibly overload the client.</td>
</tr>
<tr>
<td>Lock Database Schema</td>
<td>Specifies if the map service will acquire schema locks for map layers that come from a geodatabase. <strong>Default = True (checked)</strong></td>
<td>- Disable when you need to make schema changes to your data without stopping your services.</td>
</tr>
</tbody>
</table>
Performance Factors
Top 5 tips for good performance and scalability

1. Tune map (each scale)
2. Provide sufficient hardware resources
3. Set appropriate min and max number of instances
4. Test
5. Monitor

List: Scale-dependent rendering
   Remove unused layers
   Definition queries
   Simple symbology (Esri_Optimized)
   Annotation
   Avoid project on the fly
Provide sufficient hardware resources
Most systems are CPU bound

GIS Systems are bound by:
1. CPU - typically
2. Memory – when large number of services
3. Disk – Image Service, Synchronization
4. Network – low bandwidth deployment
5. Poorly configured virtualization can result in 30% or higher performance degradation

Most well-configured and tuned GIS systems are CPU bound.
System Tools overview

- http://www.arcgis.com
- owner:EnterpriseImp
- Show ArcGIS Desktop Content

System Tools Portal
https://systemmonitoring-emcs.esri.com/
Enterprise GIS effective monitoring
End-to-End monitoring using System Monitor

- To verify resources, must monitor end-to-end
- ArcGIS Server key stats
  - Busy Time/Transaction
  - Free instances
  - Throughput
Enterprise GIS effective monitoring
End-to-End monitoring using System Monitor

- System Monitor is available as part of an engagement with Esri Professional Services
- Available as part of an Enterprise Health Check
- Available as part of a Architecture Design activity
- Custom Rent-A-Tech activity
- VERY popular as a work plan activity available through the Esri Enterprise Advantage Program (EEAP)
  - For more information please see me after or visit us in the exhibit hall
Demo
Service Tuning and Manager
Demo
Enterprise Tools Showcase

https://systemmonitoring-emcs.esri.com/

U: esridemo p: esridemo
System Monitor – Real Example

Problem - Periodic Service Wait Timeouts, complaints of a slow response

Processor time is very low and healthy during the analysis period except for one period beginning April 10 where CPU Processor Time spiked to an anomalous 30%
System Monitor – Real Example
Problem - Periodic Service Wait Timeouts, complaints of a slow response

During the same time period, memory use was healthy and within normal to a low range. Under 4 GB of memory was committed during the monitoring period well under the 16 GB of physical memory in the machine.
Curiously, the disk idle % during the period went to 0% late on April 10 and stayed that way for hours. During this time, the disk was fully engaged with no idle time and no capacity. We expect to see disk activity fluctuate and sometimes even instantaneously reach 0%. A disk serving a production environment should not reach and sustain 0% idle for any length of time.
System Monitor – Real Example

Problem - Periodic Service Wait Timeouts, complaints of a slow response

During the time of higher than normal CPU use and little disk idle a significant amount of outbound network traffic was recorded.
What’s New?

- Single-Cluster Mode new at 10.3
  - Removes load balancing between server machines in the cluster
  - Reduces chatter and network activity

- Read-only mode new at 10.4
  - Allow you to put your GIS site into read-only mode
  - Copies a version of the config-store locally to each machine
  - Reduces network traffic
Questions

Please fill out the session survey