ArcGIS Enterprise Systems: Performance and Scalability - Testing Methodologies

Frank Pizzi
Andrew Sakowicz
Introductions

Who are we?

- Esri Professional Services, Enterprise Implementation
  - Frank Pizzi
  - Andrew Sakowicz
Introductions

- **Target audience**
  - GIS, DB, System administrators
  - Testers
  - Architects
  - Developers
  - Project managers

- **Level**
  - Intermediate
Agenda

• Performance engineering throughout project phases
• Performance Factors – Software
• Performance Factors - Hardware
• Performance Tuning
• Performance Testing
• Monitoring Enterprise GIS
Performance Engineering

Benefits

• Lower costs
  - Optimal resource utilization
  - Less hardware and licenses
  - Higher scalability

• Higher user productivity
  - Better performance
Performance Engineering throughout Project Phases
Performance Engineering throughout Project

- Tools
Performance Factors - Software
Performance Factors - Software

- Application
- GIS Services
Performance Factors - Software

- Application
  - Type (e.g., mobile, web, desktop)
  - Stateless vs. stateful (ADF)
- Design
  - Chattiness
  - Data access (feature service vs. map service)
- Output image format
Performance Factors - Software

Map service

- Performance related to number of features and vertices
Performance Factors - Software

GIS Services—Geodata

- **Database Maintenance/Design**
  - Keep versioning tree small, compress, schedule synchronizations, rebuild indexes, and have a well-defined data model.

- **Geodata Service Configuration**
  - Server Object usage timeout (set larger than 10 min. default)
  - Upload/Download default IIS size limits (200K upload/4 MB download)
Performance Factors - Software
GIS Services—Data storage

- Typically a low impact
- Small fraction (< 20%) of total response time

![Low Complexity Map: Throughput vs. data source](image)
Performance Factors - Software
GIS Services—ArcSOC instances

• ArcSOC Instances max
  - #CPU Cores (10.1)
  - #CPU Cores * n (prior to 10.1), n = 1 ... 4

Variance of Service Instances by Source Data Type
Performance Factors - Hardware
Performance Factors - Hardware

Hardware Resources

- CPU
- Network bandwidth and latency
- Memory
- Disk

Most well-configured and tuned GIS systems are processor bound.
Performance Factors - Hardware

CPU Processor Speed – Spec rate

SPEC® CINT2006 Result

Dell Inc.
PowerEdge 2950 (Intel Xeon processor X5355, 2.66 GHz)

SPECint® rate2006 = Not Run

SPECint rate base2006 = 80.9
## Design Phase—Performance Factors

### Hardware Resources—Memory

<table>
<thead>
<tr>
<th>Item</th>
<th>Low</th>
<th>High</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>XenApp Session</td>
<td>500 MB</td>
<td>1.2 GB</td>
<td>140%</td>
</tr>
<tr>
<td>Database Session</td>
<td>10 MB</td>
<td>75 MB</td>
<td>650%</td>
</tr>
<tr>
<td>Database Cache</td>
<td>200 MB</td>
<td>200 GB</td>
<td>99,900%</td>
</tr>
<tr>
<td>SOC Process (Dynamic Map Service)</td>
<td>50 MB</td>
<td>500 MB</td>
<td>900%</td>
</tr>
<tr>
<td>SOC Process (Image Service)</td>
<td>20 MB</td>
<td>1,024 MB</td>
<td>5,020%</td>
</tr>
<tr>
<td>SOC Process (Geoprocessing Service)</td>
<td>100 MB</td>
<td>2,000 MB</td>
<td>1,900%</td>
</tr>
<tr>
<td>SOM</td>
<td>30 MB</td>
<td>70 MB</td>
<td>133%</td>
</tr>
</tbody>
</table>
Performance Factors - Hardware

Performance degrades with the higher number of virtual processors.

ArcGIS for Server

Server Virtualization Efficiency

<table>
<thead>
<tr>
<th></th>
<th>Physical Server Efficiency</th>
<th>Virtualization Efficiency Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>1-3 vCPUs</td>
<td>90%</td>
<td>10%</td>
</tr>
<tr>
<td>4-6 vCPUs</td>
<td>80%</td>
<td>20%</td>
</tr>
<tr>
<td>7-8 vCPUs</td>
<td>70%</td>
<td>30%</td>
</tr>
</tbody>
</table>
Define user load first.
Performance Factors - Hardware

Network

1. Distance

2. Payload

3. Infrastructure
Performance Factors - Hardware

Hardware Resources – Network

- Impact of service and return type on network transport time
  - Compression
  - Content (e.g., Vector vs. Raster)
  - Return type (e.g., JPEG vs. PNG)

<table>
<thead>
<tr>
<th>Application Type</th>
<th>Service/Op</th>
<th>Content</th>
<th>Return Type</th>
<th>Mb/Tr</th>
<th>56 kbps</th>
<th>1.54 Mbps</th>
<th>10 Mbps</th>
<th>45 Mbps</th>
<th>100 Mbps</th>
<th>1 Gbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArcGIS Desktop</td>
<td>Map</td>
<td>Vector</td>
<td></td>
<td>10</td>
<td>0.056</td>
<td>1.540</td>
<td>10.000</td>
<td>45.000</td>
<td>100.000</td>
<td>1000.000</td>
</tr>
<tr>
<td>Citrix/ArcGIS</td>
<td>Map</td>
<td>Vector+Image</td>
<td>ICA Comp</td>
<td>1</td>
<td>1.786</td>
<td>10.000</td>
<td>1.000</td>
<td>0.222</td>
<td>0.100</td>
<td>0.010</td>
</tr>
<tr>
<td>Citrix/ArcGIS</td>
<td>Map</td>
<td>Vector</td>
<td>ICA Comp</td>
<td>0.3</td>
<td>5.357</td>
<td>0.195</td>
<td>0.030</td>
<td>0.007</td>
<td>0.003</td>
<td>0.000</td>
</tr>
<tr>
<td>ArcGIS Server</td>
<td>Map</td>
<td>Vector</td>
<td>PNG</td>
<td>1.5</td>
<td>26.786</td>
<td>0.974</td>
<td>0.150</td>
<td>0.033</td>
<td>0.015</td>
<td>0.002</td>
</tr>
<tr>
<td>ArcGIS Server</td>
<td>Image</td>
<td></td>
<td>JPG</td>
<td>0.3</td>
<td>5.357</td>
<td>0.195</td>
<td>0.030</td>
<td>0.007</td>
<td>0.003</td>
<td>0.000</td>
</tr>
<tr>
<td>ArcGIS Server</td>
<td>Map Cache</td>
<td>Vector</td>
<td>PNG</td>
<td>0.1</td>
<td>1.786</td>
<td>0.065</td>
<td>0.010</td>
<td>0.002</td>
<td>0.001</td>
<td>0.000</td>
</tr>
<tr>
<td>ArcGIS Server</td>
<td>Map Cache</td>
<td>Vector+Image</td>
<td>JPG</td>
<td>0.3</td>
<td>5.357</td>
<td>0.195</td>
<td>0.030</td>
<td>0.007</td>
<td>0.003</td>
<td>0.000</td>
</tr>
</tbody>
</table>
Network Speed Test Tool:
http://localhost/speedtest/

Demo
Performance Tuning
Tuning Process

- Profile individual user operations and tune if needed
- Drill down through software stack:
  - Application
  - Service
  - MXD
  - Layer
  - DBMS query
- *Correlate* your findings between tiers
A test is executed at the web browser. It measures web browser call’s elapsed time (round-trip between browser and data source).
Tuning

Web diagnostic tools: Fiddler

- Understand each request URL.
- Verify cache requests are from virtual directory, not dynamic map service.
- Validate host origin (reverse proxy).
- Profile each transaction response time.
Tuning
Analyze SOM/SOC statistics.

![Diagram showing the flow of data between different components: Browser, Web Server, SOM, SOC, and ArcSDE/DBMS. The diagram highlights the total response time (t1-t2), wait time, usage time, and search & retrieval time. The SOM component is circled in red, indicating a focus area for analysis.](image)
Tuning

Analyze ArcGIS for Server statistics

<Msg time="2009-03-16T12:23:22" type="INFO3" code="103021" target="Portland.MapServer" methodName="FeatureLayer.Draw" machine="myWebServer" process="2836" thread="3916" elapsed="0.05221">Executing query.</Msg>


Tuning ArcMap Publish Tool
## Tuning mxdperfstat

### Issues discovered
- Large numbers of vertices on features
- Labeling of dense features expensive

### Table of Timing Details

<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; GraphicsPhase=83;</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>Taxlot DenseLabel</td>
<td>1.84</td>
<td>Simplify labeling, symbology; GraphicsPhase=1.03; 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>1 200,001 True</td>
<td>.73</td>
<td>1.03</td>
<td>.95</td>
<td>.01</td>
<td>266</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>8,000</td>
<td>Taxlot Dense No Label</td>
<td>.53</td>
<td>simplify geometry; vertices fetched=200001;</td>
<td>1 200,001 False</td>
<td>.47</td>
<td>.02</td>
<td>.97</td>
<td>.00</td>
<td>140</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Tuning

Data Sources

- Total Response Time (t1-t2)
- Wait Time
- Usage Time
- Search & Retrieval Time

Browser

Web Server

SOM

SOC

ArcSDE/DBMS
Tuning

Data Sources—Oracle Trace

```sql
select username, sid, serial#, program, logon_time from v$session where username='STUDENT';
```

<table>
<thead>
<tr>
<th>USERNAME</th>
<th>SID</th>
<th>SERIAL#</th>
<th>PROGRAM</th>
<th>LOGON_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>---STUDENT</td>
<td>132</td>
<td>31835</td>
<td>gsrvr.exe</td>
<td>23-OCT-06</td>
</tr>
</tbody>
</table>

SQL> connect sys@gis1_andrews as sysdba

Enter password:

Connected.

SQL> execute
    sys.dbms_system.set_ev(132,31835,10046,12,'');

DBMS trace is a very powerful diagnostic tool.
### Tuning

**Data Sources—SQL Profiler**

---

**ArcSDE_trace (ANDRIW52)**

<table>
<thead>
<tr>
<th>EventClass</th>
<th>Login</th>
<th>Application</th>
<th>TextData</th>
<th>CPU</th>
<th>Duration</th>
<th>RowCounts</th>
<th>Reads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace Start</td>
<td>sde</td>
<td>SDE:5932</td>
<td>&lt;ShowPlanXML xmlns=&quot;<a href="http://schemas">http://schemas</a>....</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP:StmtCompleted</td>
<td>sde</td>
<td>SDE:5932</td>
<td>SELECT state_id,owner,creation_time...</td>
<td>10</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>SP:StmtCompleted</td>
<td>sde</td>
<td>SDE:5932</td>
<td>SELECT lineage_name, time_last_mod...</td>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>SP:StmtCompleted</td>
<td>sde</td>
<td>SDE:5932</td>
<td>&lt;ShowPlanXML xmlns=&quot;<a href="http://schemas">http://schemas</a>....</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SP:StmtCompleted</td>
<td>sde</td>
<td>SDE:5932</td>
<td>SELECT S_.eminy,S_.eminy,S_.emaxy,S...</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SP:StmtCompleted</td>
<td>sde</td>
<td>SDE:5932</td>
<td>&lt;ShowPlanXML xmlns=&quot;<a href="http://schemas">http://schemas</a>....</td>
<td>521</td>
<td>2624</td>
<td>38251</td>
<td>11</td>
</tr>
</tbody>
</table>

---

**Nested Loop (Inner Join)**

- Cost: 0%

- Sorted!

- Distinct Sort: 34%

- Clustered Index Seek:
  - [SDE], [SDE], [F2], [F2.pk] [SDE] Index
  - Cost: 5%

---

**Index Scan**

Scan a nonclustered index, entirely or only a range.

**Physical Operation**

- Index Scan

**Logical Operation**

- Index Scan

- Actual Number of Rows: 51629
- Estimated I/O Cost: 1.1142
- Estimated CPU Cost: 0.183391
- Estimated Operator Cost: 1.29763 (63%)
- Estimated Subtree Cost: 1.29760
- Estimated Number of Rows: 21.77
- Estimated Row Size: 99
- Actual Rebinds: 0
- Actual Rebinds: 0
- Ordered: False
- Node ID: 5
Performance Testing
Testing
Performance Testing—Objectives

- Define Objectives
  - Contractual Service-Level Agreement?
- Bottlenecks
- Capacity
- Benchmark
Testing
Performance Testing—Prerequisites

- Functional testing completed
- Performance tuning
Testing

Performance Testing—Test Plan

- Test Plan
  - Workflows
    - Expected User Experience (Pass/Fail Criteria)
    - Single User Performance Evaluation (Baseline)
    - Think Times
    - Active User Load
    - Pacing
    - Valid Test Data and Test Areas
  - Testing Environment
    - Scalability/Stability
    - IT Standards and Constraints
    - Configuration (GIS and Non-GIS)
Testing

Performance Testing—Test tools

- Tool selection depends on objective.
  - Commercial tools all have system metrics and correlation tools.
  - Free tools typically provide response times and throughput but leave system metrics to the tester to gather and report on.
<table>
<thead>
<tr>
<th>Test Tools</th>
<th>Open Source</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>LoadRunner</td>
<td>No</td>
<td>• Industry Leader</td>
<td>• High cost</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Automatic negative correlations identified with service-level agreements</td>
<td>• Test development in C programming language</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• HTTP web testing</td>
<td>• Test metrics difficult to manage and correlate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Click and script</td>
<td>• Poor user community with few available examples</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Very good tools for testing SOA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Test results stored in database</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Thick client testing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Can be used for bottleneck analysis</td>
<td></td>
</tr>
<tr>
<td>Silk Performer</td>
<td>No</td>
<td>• Good solution for testing Citrix</td>
<td>• Moderate to high cost</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Wizard-driven interface guides the user</td>
<td>• Test metrics are poor.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Can be used for bottleneck analysis</td>
<td>• Test development uses proprietary language.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Test metrics difficult to manage and correlate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Poor user community with few available examples</td>
</tr>
<tr>
<td>Visual Studio Test Team</td>
<td>No</td>
<td>• Low to moderate cost</td>
<td>• No built-in support for AMF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Excellent test metric reporting</td>
<td>• No thick-client options</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Test scripting in C# or VB .NET</td>
<td>• Moderate user community</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Unit and web testing available</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Blog support with good examples</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Very good for bottleneck analysis</td>
<td></td>
</tr>
<tr>
<td>JMeter</td>
<td>Yes</td>
<td>• Free</td>
<td>• Provides only response times</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tool</td>
<td>• Poor user community with few available examples</td>
</tr>
</tbody>
</table>
Testing

Test Data – heat map

Observe correlation between feature density and performance.
Testing

Load Test

• Create load test.
  - Define user load.
    - Max users
    - Step interval and duration
  - Create machine counters to gather raw data for analysis.

• Execute.
Testing

Execute

• Ensure
  - Virus scan is off
  - Only target applications are running
  - Application data is in the same state for every test
  - Good configuration management is critical to getting consistent load test results
Demo: System Test
Performance and Scalability

Definitions

- Performance: The speed at which a given operation occurs
- Scalability: The ability to maintain performance as load increases
Load Test

Throughput (request/hr)

Step Load (users)

Throughput(req/hr)

Response Time (sec)

~85% Utilization

time
Load Test

Resource utilization: CPU, Memory, Network

- Throughput (req/hr)
- CPU Utilization (%)
- Network used (Mbps)
- Response Time (sec)
- Memory used (Mb)

Graph showing the impact of Step Load (users) on system resources over time.
Load Test

Validation: stats correlation, steady content length, failed requests=0
Testing

- Analysis—Compare and correlate key measurements

- Most counters and utilization should be increasing with increased load:
  - Throughput
  - Response time
  - Metrics
    - CPU
    - Network
    - Disk
    - Memory
  - Errors
Load Test

System capacity

- Step Load (users)
- Throughput (req/hr)
- CPU Utilization (%)
- Network used (Mbps)
- Content length (bytes)
- Memory used (Mb)
- Response Time (sec)

~85% Utilization
Load Test
System capacity

- System capacity can be defined as a user load corresponding to
  - Maximum throughput
  - Threshold utilization (e.g., 80)
  - SLA response time
Testing
Analysis—Valid range

- Exclude failure range (e.g., failure rate > 5%) from the analysis.
- Exclude excessive resource utilization range.
Testing
Analysis—Compare and correlate key measurements

- **Unexpected curve**

![Graph showing unexpected curved shape in response time vs user load. The curve is not as expected, where response time should be increasing. Likely root cause: failed or 0 size image requests.](image)
Development Phase—Testing

Analysis—Compare and correlate key measurements

- Expected counters correlation: increasing user load, CPU utilization, response time
Testing
Analysis—Compare and correlate key measurements

• Memory leak example

Root cause:
Web Server process
Test Results as Input into Capacity Planning

Load Test Results—Riverside Electric

- Baseline Test with Single Thread
  - Note* Service Time Is Load Independent
- Think Time=0
- Evaluate Key Metrics
  - Throughput
  - Response Time
  - QA Check
- Evaluate System Under Test.
  - CPU, Network, Memory, and Disk
Test Results as Input into Capacity Planning

Load Test Results—Key Indicators
Test Results as Input into Capacity Planning

Load Test Results—System Metrics
Test Results as Input into Capacity Planning

Load Test Results – Input into capacity models

- Throughput = 3.89 request/sec ~ 14,004 request/hour
- CPU Utilization = 20.8%
- Mb/request = 1.25 Mb
Test Results as Input into Capacity Planning

Load Test Results – Input into CPU capacity model

- **Input from testing**
  - #CPUs = 4 cores
  - %CPU = 20.8
  - TH = 14,004 requests/hour
  - SPEC per Core of machine tested = 35

- **ST = \((4 \times 3600 \times 20.8) / (14,004 \times 100)\) = 0.2138 sec**
  - Note: Very close to Average response time of 0.25
Test Results as Input into Capacity Planning

Target values

1. Server SpecRate/core = 10.1

![SPEC® CINT2006 Result](image)

2. User load = 30,000 req/hr

3. Network = 45 Mbps
Test Results as Input into Capacity Planning

Target CPU cores calculation

• Input to Capacity Planning:
  - ST = Service Time = 0.2138 sec
  - TH = Throughput desired = 30,000 requests/hour
  - %CPU = Max CPU Utilization = 80%
  - SpecRatePerCpuBase = 35
  - SpecRatePerCpuTarget = 10.1

• Output
  - #CPU required = ( [.2138*30,000*100]/3600*80) *[35/10.1]
  - #CPU required = 7.7 cores ~ 8 cores
Test Results as Input into Capacity Planning

Target network calculation

- **Input to Capacity Planning:**
  - \( Mb/\text{req} = 1.25 \)
  - \( TH = 30,000 \text{ requests/hour} \)

- **Output**
  - Network bandwidth required = \( 30000 \times 1.25 / 3600 \)
  - = 10.4 Mbps < 45 Mbps available
  - Transport = \( 1.25 / (45 - 10.4) = 0.036 \text{sec} \)

\[
Mbps = \frac{TH \times Mbits / \text{req}}{3600}
\]

\[
\text{Transport(sec)} = \frac{Mbits / \text{req}}{Mbps - Mbps_{\text{used}}}
\]
Test Results as Input into Capacity Planning

System Test Report

Environment Performance and Throughput Vs. Step Load
Monitoring Enterprise GIS

- **Hardware (perfmon)**
  - CPU, Memory, Network, Disk
  - Uptime
- **GIS Services (ArcGIS for Server Stats, IIS logs)**
  - Response Time
  - Throughput
  - Uptime
- **Geodatabase (ArcPy)**
  - Number of connections
  - Feature class count
- **DBMS (system views)**
Demo: System Monitor
Contact Us

- Frank Pizzi  
  - fpizzi@esri.com
- James Livingston  
  - jlivingston@esri.com
- Aaron Lopez  
  - alopez@esri.com
- Andrew Sakowicz  
  - asakowicz@esri.com
Performance Engineering throughout Project

- Tools

System Tools - Capacity Planning

System Designer

System Monitor
- User Load
- CPU%

System Test
- Capacity models
Download Tools

- Open Windows Explorer (not browser).
- Right-click and select Login As (or click Alt F and select Login As from the File).
- Enter your user name and password:
  - **User name:** eist
  - **Password:** eXwJkh9N
- Click Log On.
- Follow **Installation Guide**.
- Report bugs and provide feedback:
  - SystemDesigner@esri.com
• Thank you for attending
• Open for Questions

Frank Pizzi, fpizzi@esri.com
Andrew Sakowicz, asakowicz@esri.com

• Please fill out the evaluation:

www.esri.com/ucsessionssurveys

Offering ID: 983