ArcGIS for Server Performance and Scalability—Testing Methodologies

Andrew Sakowicz
Frank Pizzi
Introduction

- Esri Professional Services
- Frank Pizzi, fpizzi@esri.com
- Andrew Sakowicz, asakowicz@esri.com
Audience

- Testers
- Developers
- Architects

Please!

*Turn OFF cell phones and paging devices*
Outline

- Performance factors
- Tuning and monitoring
- Testing
- Capacity planning
Performance factors
Performance engineering

Addressed through each project phases

• Testing and tuning conducted typically during:
  • Development
    - Prototype
    - Unit test
  • Deployment
  • Operations
Performance and Scalability

Definitions

- **Performance**: The speed at which a given operation occurs
- **Scalability**: The ability to maintain performance as load increases
GIS Services

Map service

- Performance related to number of features and vertices
Most well-configured and tuned GIS systems are processor-bound.
Virtualization overhead

10% to 30%
Network transport time

1. Distance
   ![Diagram of distance](image1)

2. Payload
   ![Diagram of payload](image2)

3. Infrastructure
   ![Diagram of infrastructure](image3)
Network transport time

\[
\text{Transport (sec)} = \frac{\text{MbpTr}}{\text{Mbps} - \text{Mbps}_{\text{used}}}
\]
Network transport time

- 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>0.056</td>
<td>1.540</td>
<td>10.000</td>
<td>45.000</td>
<td>100.000</td>
<td>1000.000</td>
<td>0.010</td>
</tr>
<tr>
<td>Citrix/ArcGIS</td>
<td>Map</td>
<td>Vector+Image</td>
<td>ICA Comp</td>
<td>1</td>
<td>17.857</td>
<td>0.649</td>
<td>0.100</td>
<td>0.022</td>
<td>0.010</td>
<td>0.001</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>JPG</td>
<td></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>
Demo: Network speed test

System Monitor Speed Test

Download speed: 11.99 Mb/s
3.32 s

Payload: bigfile.bin size: 4996 Kb
Iterations: 10

Speed Test

Graph showing speed over iterations.
## Memory

<table>
<thead>
<tr>
<th>Item</th>
<th>Low</th>
<th>High</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Map</td>
<td>50 MB</td>
<td>500 MB</td>
<td>900%</td>
</tr>
<tr>
<td>Image Service</td>
<td>20 MB</td>
<td>1,024 MB</td>
<td>5,020%</td>
</tr>
<tr>
<td>Geoprocessing</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>
<tr>
<td>XenApp Session</td>
<td>500 MB</td>
<td>1.2 GB</td>
<td>140%</td>
</tr>
<tr>
<td>DBMS Session</td>
<td>10 MB</td>
<td>75 MB</td>
<td>650%</td>
</tr>
<tr>
<td>DBMS Cache</td>
<td>200 MB</td>
<td>200 GB</td>
<td>99,900%</td>
</tr>
</tbody>
</table>
User load

User has the highest uncertainty

![Bar chart showing uncertainly levels for different categories: Active Users, Think Time, Capacity Model, Operation Details, and Hardware (SpecRate).](chart.png)
Tuning and Monitoring
Tuning process

1. Profile and measure response time at the client application
2. Conduct measurements at software stack below
3. Correlate and account measurements between tiers
4. Identify root cause

*Do not misdiagnose “victims for culprits”*
Measure response time at the client application

A test is executed at the web browser. It measures web browser call’s elapsed time (roundtrip between browser and data source).
Demo: Profile web application

Fiddler http://www.fiddler2.com/
Analyze ArcGIS Server statistics using Arc Catalog, Manager or logs.
Analyze ArcGIS Server statistics

Correlate and account measurements between tiers

Executable query.

Feature count: 27590

End of layer draw: STREETS
Web server log

Log Parser

d06b-abf8-4c25-91b2-f8d975cf8c07&displaylang=en)

Logparser "SELECT date, QUANTIZE(time, 3600) as Hour, cs-uri-stem, count(*) as Req/hr

FROM C:\inetpub\logs\LogFiles\W3SVC1\u_ex120308.log

WHERE cs-uri-stem like

'%/arcgis/rest/services/World_Street_Map_MapServer1/MapServer/export%'
group by date, Hour, cs-uri-stem order by Hour"
ArcGIS Server log

ASLog

Identify root cause

Analyze Map Tool
Identify root cause

Mxdperfstat on http://resources.arcgis.com

C:>mxdperfstat -mxd Portland_Dev09_Bad.mxd -xy 7655029;652614 -scale 8000

<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>TaxlotDenseLabel</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</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>
Analyze database statistics

Correlate and account measurements between tiers

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

**Browser**

**Web Server**

**SOM**

**SO C**

**SDE/DBMS**

Analyze database statistics and correlate and account measurements between tiers.
Analyze database statistics

Oracle Trace

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

```
USERNAME     SID    SERIAL# PROGRAM    LOGON_TIME
------------------------------ ---------- ---------- ----------------------------
--------STUDENT   132      31835 gsrvr.exe    23-OCT-06
```

```
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.*
Analyze database statistics

SQL Profiler
Testing
Testing

Objectives

• Capacity planning
• Resource bottlenecks
• Benchmark

Iterative process
Testing steps

- Validated
  - functional
  - configuration
  - performance of single user operation
- Create a test data
- Develop test scripts, including user load
- Run test, including monitoring
- Analyze results
Validate configuration

- Critical step in successful GIS deployments
- Incorrect settings can limit access to hardware resources
- Configure ArcGIS Server Instances
Testing

Test representative area of interests

• Is Random data appropriate?
Test data

Test representative area of interests

- **PerfHeatMap**
  - Red – slowest
  - Green - fastest

Testing the ArcGIS Server REST API
Test data

- **Bounding box**
  - Use representative areas of interest

- **Attribute Data**
  - Can be used to parameterize a web request

- **Geometry Data**
  - Points
  - Lines
  - Polygons
**Test data**

**Bbox data**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dsl</td>
<td>96</td>
</tr>
<tr>
<td>transparent</td>
<td>true</td>
</tr>
<tr>
<td>format</td>
<td>png8</td>
</tr>
<tr>
<td>bbox</td>
<td>-21.3755123871000073,26.837690194625327,81.54447185264456,89.97625310033272</td>
</tr>
<tr>
<td>bboxSR</td>
<td>4326</td>
</tr>
<tr>
<td>imageSR</td>
<td>4326</td>
</tr>
<tr>
<td>size</td>
<td>1247,765</td>
</tr>
<tr>
<td>F</td>
<td>image</td>
</tr>
</tbody>
</table>

![Map of Europe with a legend indicating data regions](image-url)
Test scripts

- Record user workflow
- Create single user web test
  - Define transactions
  - Set think time and pacing
  - Parameterize transaction inputs using test data
  - Verify test script with single user
Load test

- Create load test
  - User load
  - Performance counters
Load test

Avoid these mistakes

- Applying unreasonable load
- Running too many tests
- Deployment is not configured correctly
- Deployment is not exclusive to testing
- Test results are not repeatable
- Test client is bottleneck
- Test definition does not have proper validation rules
Analyze results

• Compare and correlate key measurements
  - Response Time Vs. Throughput
  - CPU, Network, Disk, and Memory on all tiers
  - Passed and Failed tests

• Validation
  - Lack of errors does not validate a test
  - Spot check request response content size
Analyze results

Valid

- *Expected* CPU and Response time correlation
Analyze results

Invalid

• Validation Example
  - *Unexpected* response time under heavy load?

![Graph showing unexpected curve shape in response time vs user load](image-url)
Analyze results

Invalid

- **Validation Example**
  - Test failure – memory bottleneck in w3wp process
Analyze results

Invalid

- Validation Example – *Unexpected* CPU utilization
Test Case: Accounting for database CPU
Test Case: Accounting for database CPU

Profile a single user – record cpu

WORKLOAD REPOSITORY report for

<table>
<thead>
<tr>
<th>Host Name</th>
<th>Platform</th>
<th>CPUs</th>
<th>Cores</th>
<th>Sockets</th>
<th>Memory (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvtlinux.esri.com</td>
<td>Linux x86 64-bit</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>7.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Snap Id</th>
<th>Snap Time</th>
<th>Sessions</th>
<th>Cursors/Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>Begin Snap:</td>
<td>3008</td>
<td>32</td>
<td>8.2</td>
</tr>
<tr>
<td>End Snap:</td>
<td>3010</td>
<td>35</td>
<td>12.3</td>
</tr>
<tr>
<td>Elapsed:</td>
<td>16.18 (mins)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DB Time:</td>
<td>1.29 (mins)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Host CPU (CPUs: 4 Cores: 4 Sockets: 2)

<table>
<thead>
<tr>
<th>Load Average Begin</th>
<th>Load Average End</th>
<th>%User</th>
<th>%System</th>
<th>%WIO</th>
<th>%Idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>0.05</td>
<td>1.7</td>
<td>0.1</td>
<td>1.7</td>
<td>98.1</td>
</tr>
</tbody>
</table>

Instance CPU

<table>
<thead>
<tr>
<th>%Total CPU</th>
<th>%Busy CPU</th>
<th>%DB time waiting for CPU (ResourceManager)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.7</td>
<td>91.5</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Test Case: Accounting for database CPU

Profile a single user – find top queries

WORKLOAD REPOSITORY report for

SQL ordered by CPU Time
Test Case: Accounting for database CPU

Validate load test

CPU Utilization
users=20, think time=0

(expected CPU Utilization=1.7% * 20 users = 34%)
Executive Summary
Test Plan
  - Workflows and Work load
Deployment documentation
Results and Charts
  - Key Indicators, e.g. Response Time, Throughput
  - System Metrics, e.g. CPU %
  - Errors
Summary and Conclusions
Appendix
• Determine System Capacity
  - Resource Utilization > X%
  - Maximum acceptable response times
    - 95% of transactions under X seconds
    - Max response time < X seconds
  - Identify bounding factor for each tier
    - Document capacity for each tier component
    - Document bounding factor for each tier
Testing

Test Tools

• Commercial Tools
  - Load Runner
  - Visual Studio
  - Silk Performer

• Free Tools
  - Apache JMeter
  - Open STA
  - WCAT (Fiddler extension simplifies use)
  - SoapUI
  - Curl
Using test results as input for capacity planning
Throughput (Req/sec)
CPU utilization
Test Results as Input into Capacity Planning

Load Test Results – input into capacity models

- **Average throughput over the test duration**
  - 3.89 request/sec ~ 14,004 request/hour

- **Average response time over the test duration**
  - .25 seconds

- **Average 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 = \( \frac{4 \times 3600 \times 20.8}{14,004 \times 100} \) = 0.2138 sec
- Note* very close to Average response time of .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:**
  - \( ST = \text{Service Time} = .2138 \text{ sec} \)
  - \( TH = \text{Throughput desired} = 30,000 \text{ request/hour} \)
  - \( \%CPU = \text{Max CPU Utilization} = 80\% \)
  - \( \text{SpecRatePerCpuBase} = 35 \)
  - \( \text{SpecRatePerCpuTarget} = 10.1 \)

- **Output**
  - \( \#CPU \text{ required} = \left( \frac{.2138 \times 30,000 \times 100}{3600 \times 80} \right) \left( \frac{35}{10.1} \right) \)
  - \( \#CPU \text{ required} = 7.7 \text{ cores} \sim 8 \text{ cores} \)

\[
\# CPU_t = \frac{ST_b \times TH_t \times 100}{3600 \times \%CPU_t} \times \frac{\text{SpecRatePerCpuBase}}{\text{SpecRatePerCpuTarget}}
\]
Test Results as Input into Capacity Planning

Target network calculation

• Input to Capacity Planning:
  - Mb/req=1.25
  - TH = 30,000 request/hour

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

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

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

System Designer

- **Input:**
  - Throughput=30000
  - ST=0.21
  - Mb/tr=1.25
  - Hardware=80.9 Spec
Test Results as Input into Capacity Planning

System Designer

- Input
  - Hardware=80.9 Spec
Test Results as Input into Capacity Planning

System Designer

- Review results
Design Tools

System Designer

- Gathering requirements
- Designing
- Capacity: CPU, Network, Memory
- Reporting
Design Tools

System Designer Templates
System Designer

• Download from:
  - Open Explorer
  - In the Address Bar enter: ftp://ftp.esri.com/
  - Right click or select the File menu and choose Login As
  - Enter your username and password:
    - username: eist
    - password: eXwJkh9N
    - Click "Log On"

• Contact
  - asakowicz@esri.com
Questions?

Contact us:
Frank Pizzi, fpizzi@esri.com
Andrew Sakowicz, asakowicz@esri.com