Esri Petroleum User Group Conference

April 18–21, 2011 • Houston, Texas

ArcGIS Server Performance and Scalability – Optimization and Testing

Andrew Sakowicz

Objective

- **Overview:**
  - **Key performance factors**
  - **Optimization techniques**
  - **Performance Testing**

Introduction

- Andrew Sakowicz
  - Esri Professional Services, Redlands
  - asakowicz@esri.com

Audience

- Audience
  - Developers
  - Architects
  - GIS Administrators
  - DBA’s
- Level:
  - Intermediate

Performance factors

Performance Factors: ArcGIS Server Services

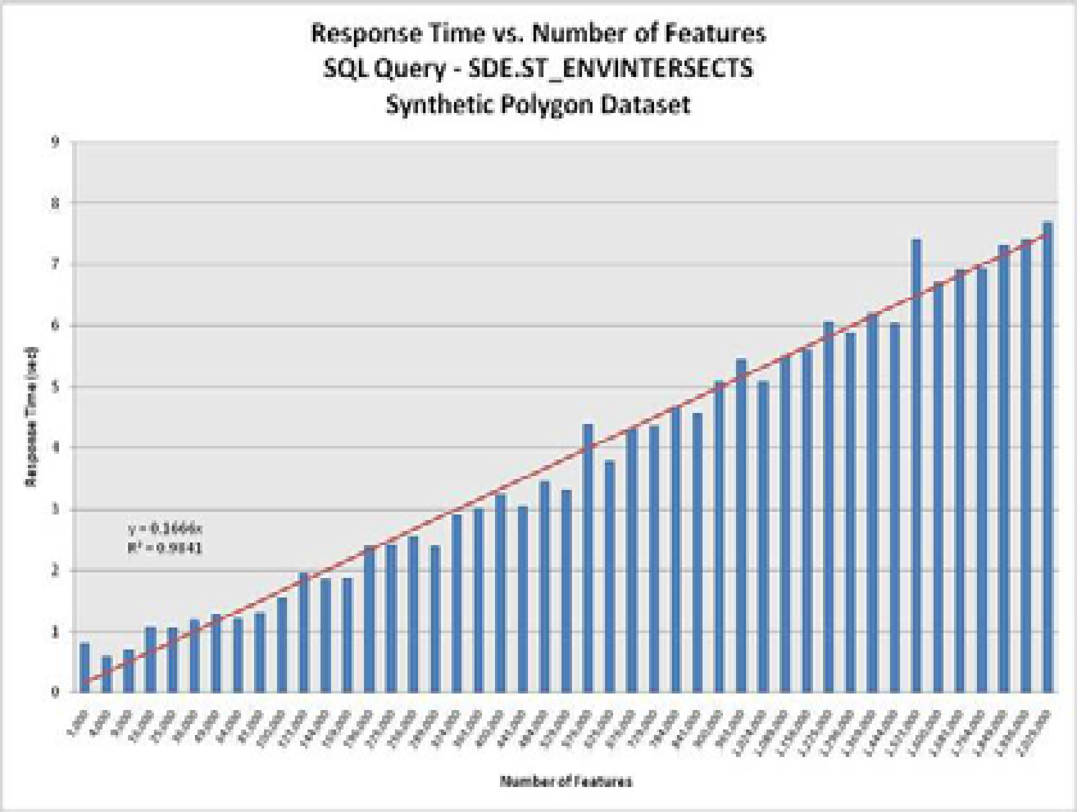
Map Service – Source document optimization

- Keep map symbols simple
- Scale dependency
- Optimize spatial index
- Simplify data
- Avoid re-projections on the fly
- Optimize map text and labels for performance
- Use annotations
- Avoid wavelet compression-based raster types (MrSid,JPEG2000)
- Use fast joins (no cross db joins)

Performance Factors: ArcGIS Server Services

Map Service – Source map document optimizations, scale dependency

- **Performance linearly related to number of features**



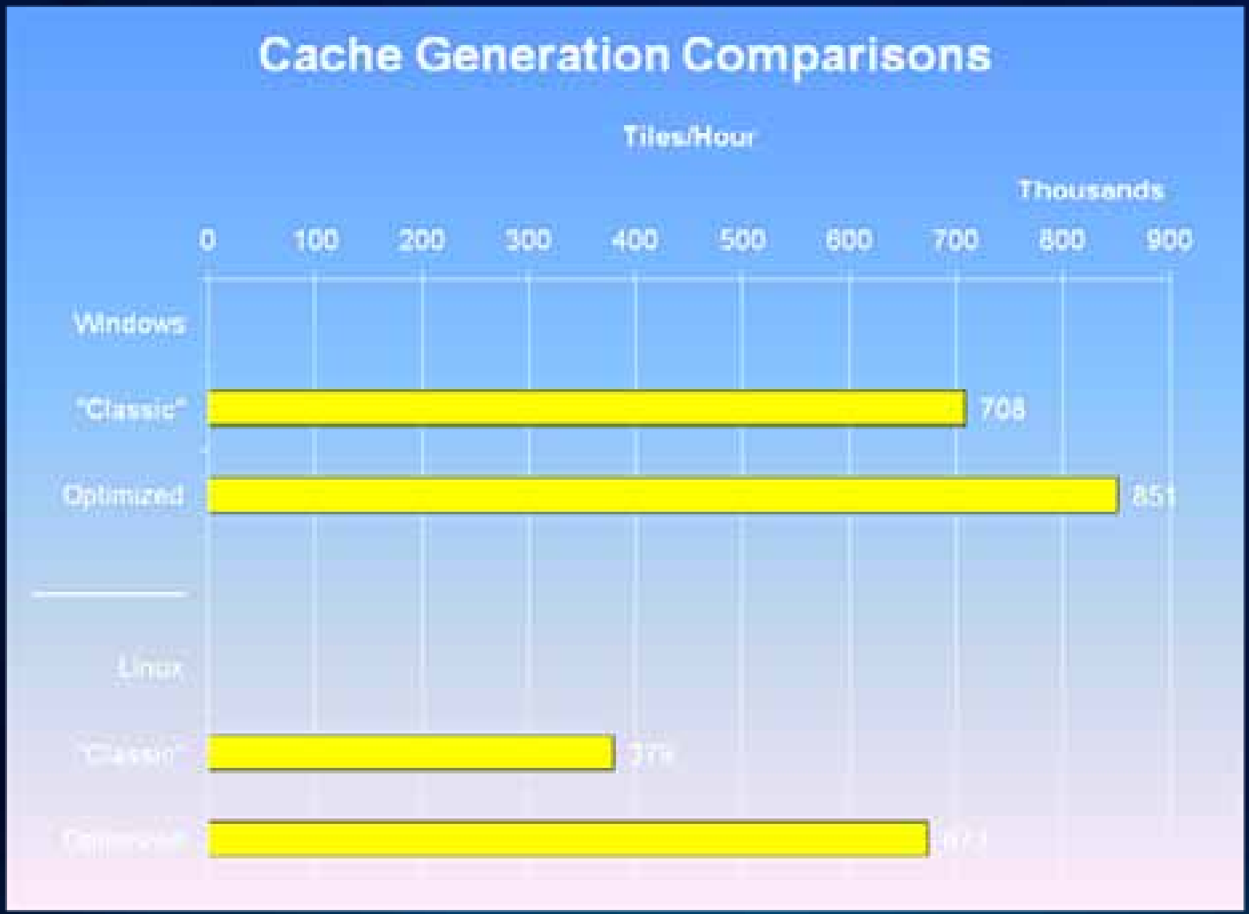
Performance Factors: ArcGIS Server Services

Map Service – Output image format choices

- PNG8/24/32
  - Transparency support
  - 24/32 good for anti-aliasing, rasters with many colors
  - Lossless: Larger files (> disk space/bandwidth, longer downloads)
- JPEG
  - Basemap layers (no transparency support)
  - Much smaller files

Performance Factors: ArcGIS Server Services

- Optimized services create caches significantly faster than “Classic



Performance Factors: ArcGIS Server Services

Geoprocessing Service

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

Detailed instructions on the Resource Center

Performance Factors: ArcGIS Server Services

Image Service

- Tiled, JPEG compressed TIFF is the best (10-400% faster)
- Build pyramids for raster datasets and overviews for mosaic datasets
- Tune mosaic dataset spatial index.
- Use JPGPNG request format in Web and Desktop clients
  - Returns JPEG unless there are transparent pixels (best of both worlds).

Help Topic: “Optimization Considerations for ArcGIS Image Server”

Performance Factors: ArcGIS Server Services

Geocode Service

- Use local instead of UNC locator files.
- Services with large locators take a few minutes to “warm-up”
- New 10.0 Single Line Locators offer simplicity in address queries but might be slower than traditional point locators.

Performance Factors: ArcGIS Server Services\

Mobile Service

- Document Preparation
  - Minimize operational layers
  - Cache basemap layers
- Service Configuration
  - Try to keep total service cache size under 250 MB
- Usage considerations
  - Avoid batch postings in favor of frequent updates

Performance Factors: ArcGIS Server Services

Feature/Geodata Service – Database maintenance is key

- **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/4MB download)**
- **Feature Service**
  - **Trade-off between client-side rendering and sending large amounts of data over the wire.**

Performance Factors: CPU Type

Select adequate hardware to support desired performance/load

- **CPU**
  - **Select for intended use**
    - **Mapping: highest Baseline CINT Rate/Core**
    - **GP: highest Baseline CFP Rate/Core**
  - **Sizing**
    - **Published CPU benchmarks:** *http://www.spec.org/cpu2006/results/cint2006.html*
    - **Published CPU-limited ESRI benchmarks:** *http://resources.esri.com/enterprisegis/index.cfm?fa=codeGallery*

Performance Factors: Hardware Resources

Ensure sufficient CPU, Memory and Network resources

- **User load: concurrent users or throughput**
- **Operation CPU service time (model) - performance**
- **CPU type**

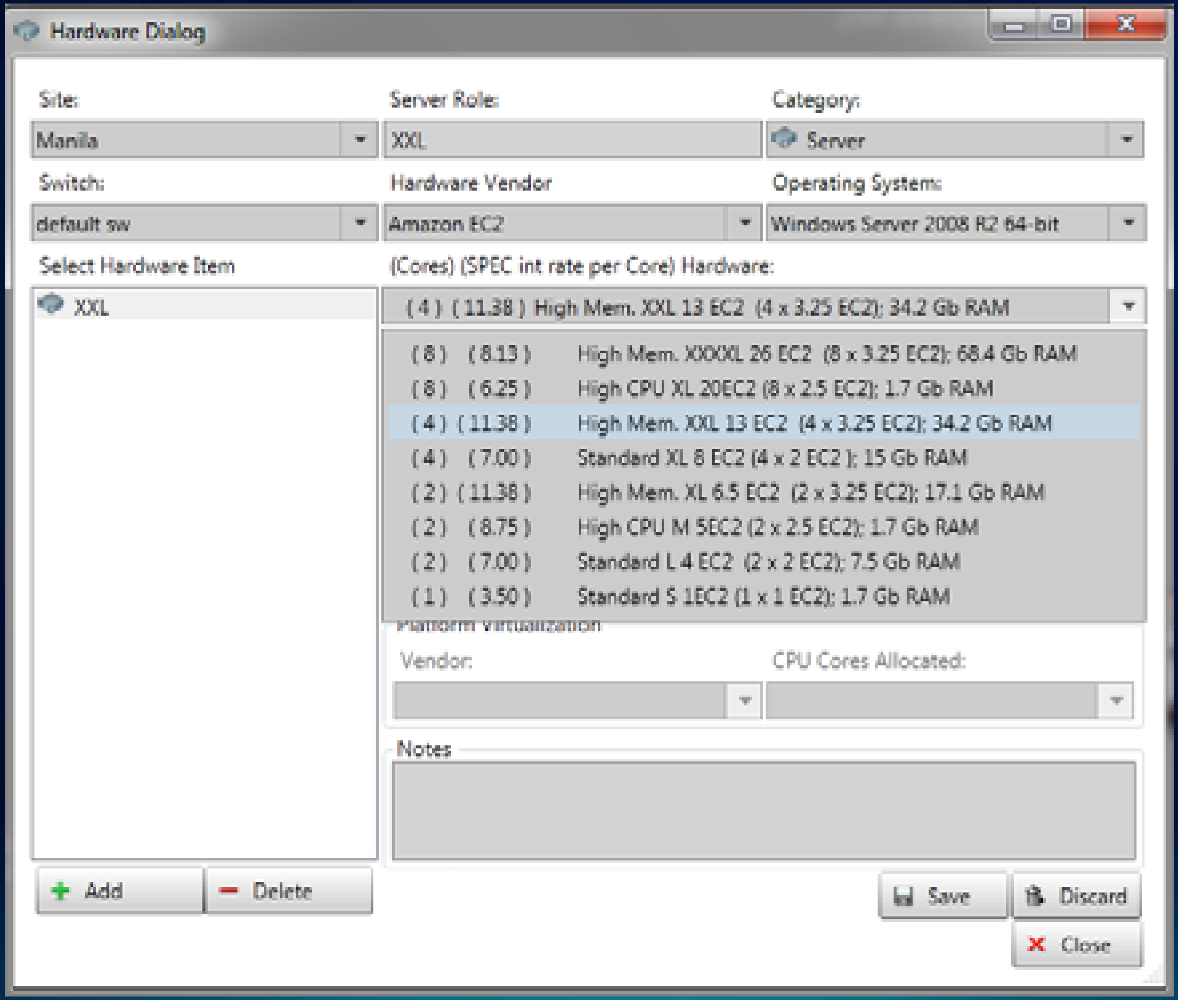
$$\#CPU_t = \frac{ST_b \times TH_t \times 100}{3600 \times \%CPU_t} \times \frac{SpecRatePerCPU_b}{SpecRatePerCPU_t}$$

subscript t = target
subscript b = benchmark
ST = CPU service time
TH = throughput
%CPU = percent CPU

Performance Factors: CPU Type

Select adequate hardware to support desired performance/load

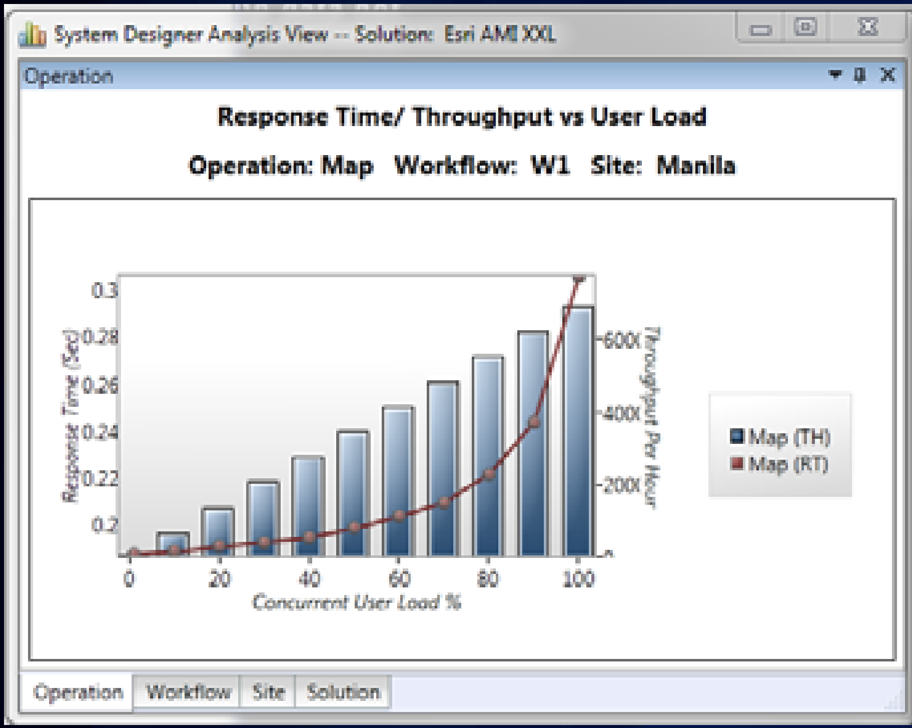
- **System Designer**

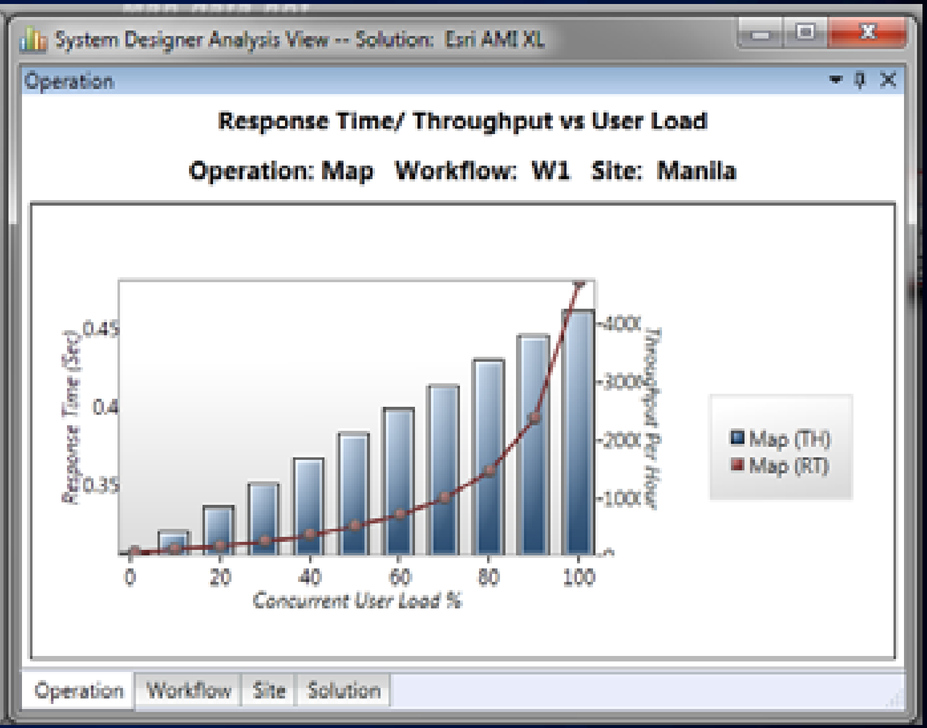


Performance Factors: CPU Type

CPU type impacts response time (performance) and capacity (scalability)

- **Amazon instance: XXL vs. XL**

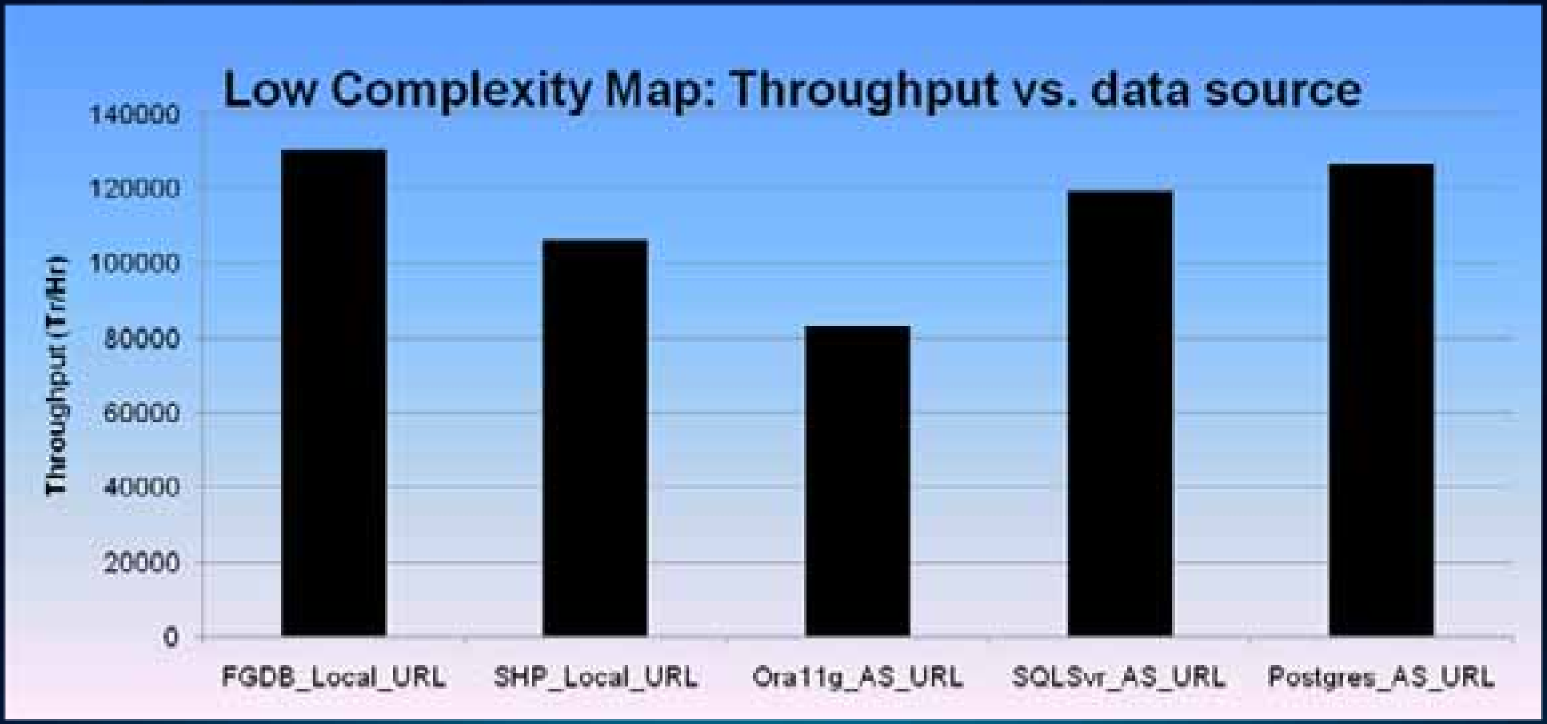




Performance Factors: Data Sources

Select data storage format that provides optimal performance

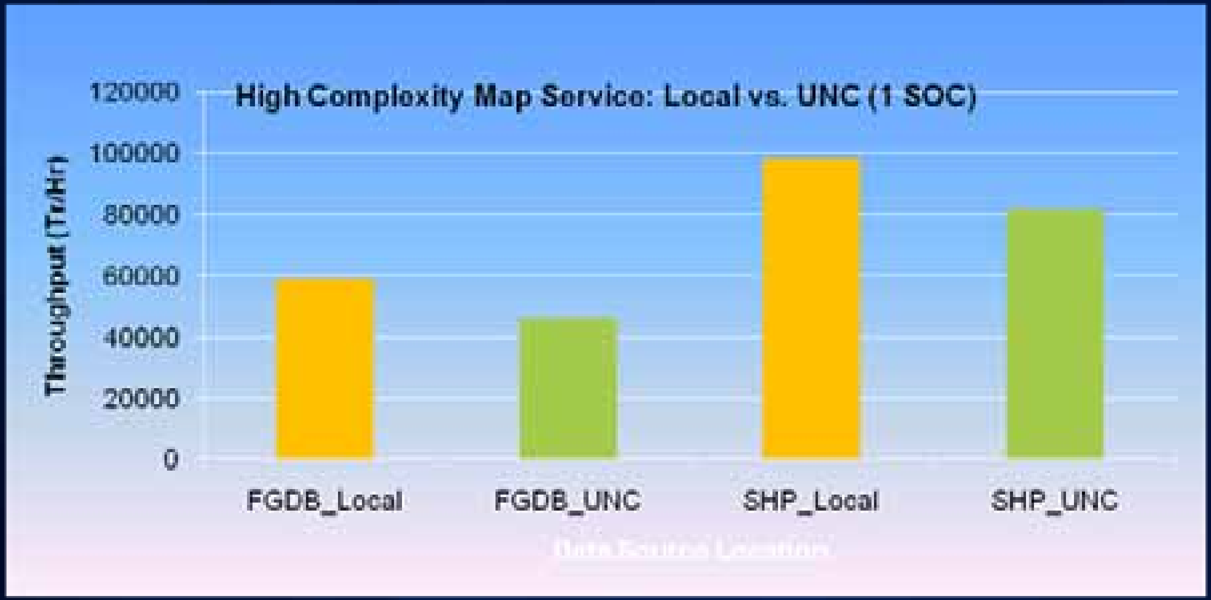
- **Data storage format**
  - RDBMS, FileGDB, Shapefile, SDC



Performance Factors: Data Location

Select data location that provides optimal performance

- Local to SOC machine
- UNC (protocol + network latency/bandwidth penalties)



- **All disks being equal, locally sourced data results in better throughput.**

Performance Factors: DB Management

Optimize DB configuration and conduct maintenance

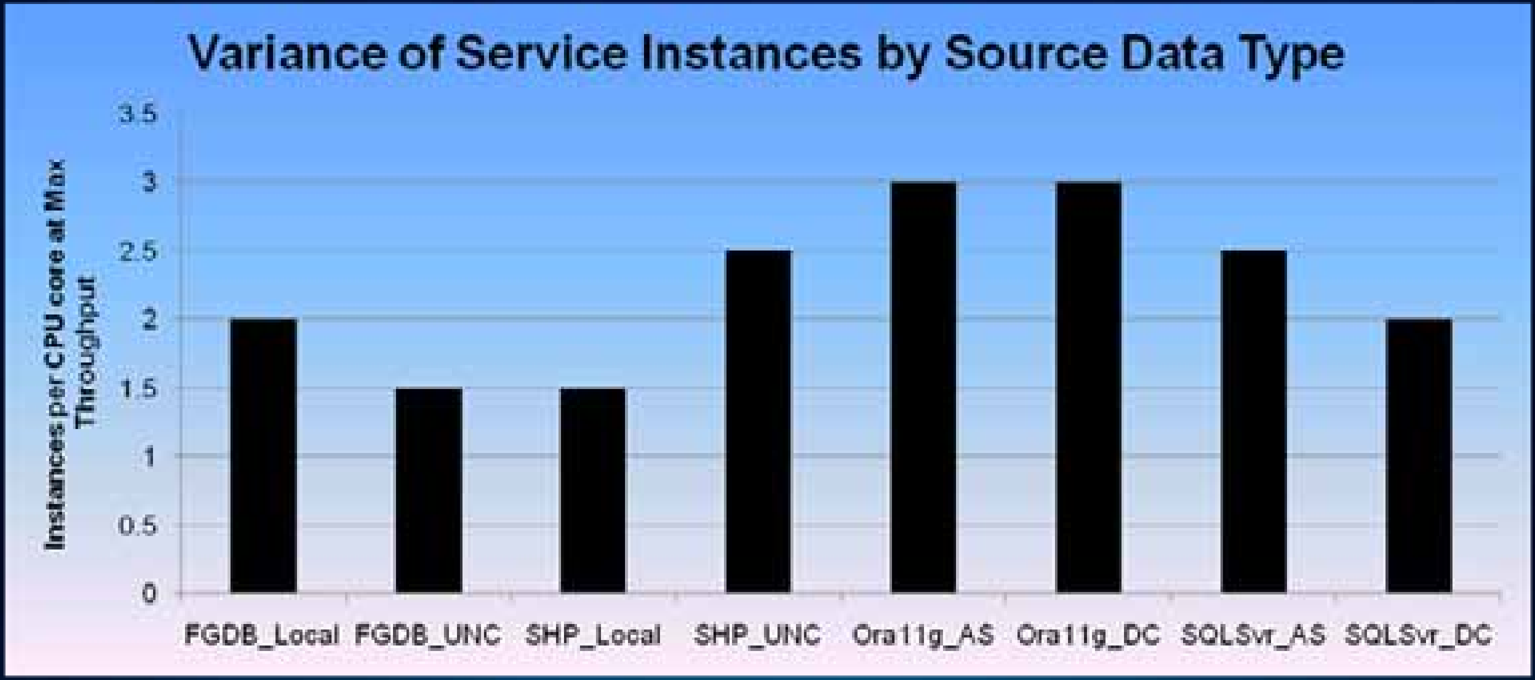
- **DBMS configuration**
- **Create and maintain (rebuild) attribute indexes**
- **Updating DBMS statistics**
- **Versioning management**
  - **Reconcile and post**
  - **Compress**

Non- optimal DBMS may be a source of significant performance degradation

Performance Factors: ArcGIS Server Framework

SOC

Optimal number of instances/core departs from CPU-limited value of 1 by choice of source data type/location.



Tuning primer

Tuning Steps

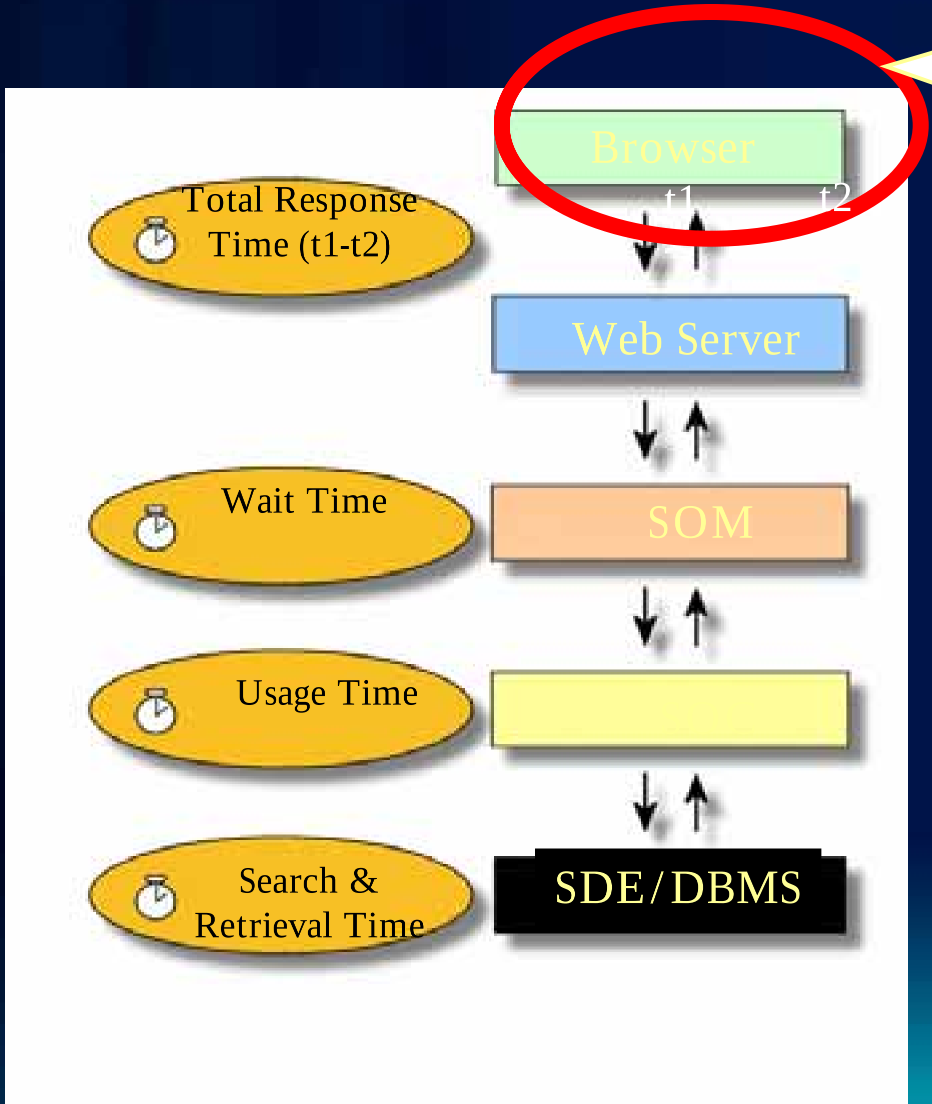
- Optimize ArcGIS Services
- Profile individual user operations and tune if needed
- Drill down through software stack
  - Application
  - Service
  - Mxd
  - Layer
  - DBMS query
- Correlate your findings between tiers
- Performance and load test

Performance tuning

- Benefits
  - Improved performance - user experience
  - Optimal resource utilization – scalability
- Tools
  - Fiddler
  - Mxdperfstat, http://resources.arcgis.com/gallery/file/enterprise-gis/details?entryID=6391E988-1422-2418-88DE-3E052E78213C
  - Map Service Publishing Toolbar
  - DBMS trace

Tuning Primer

Profile user transaction response time

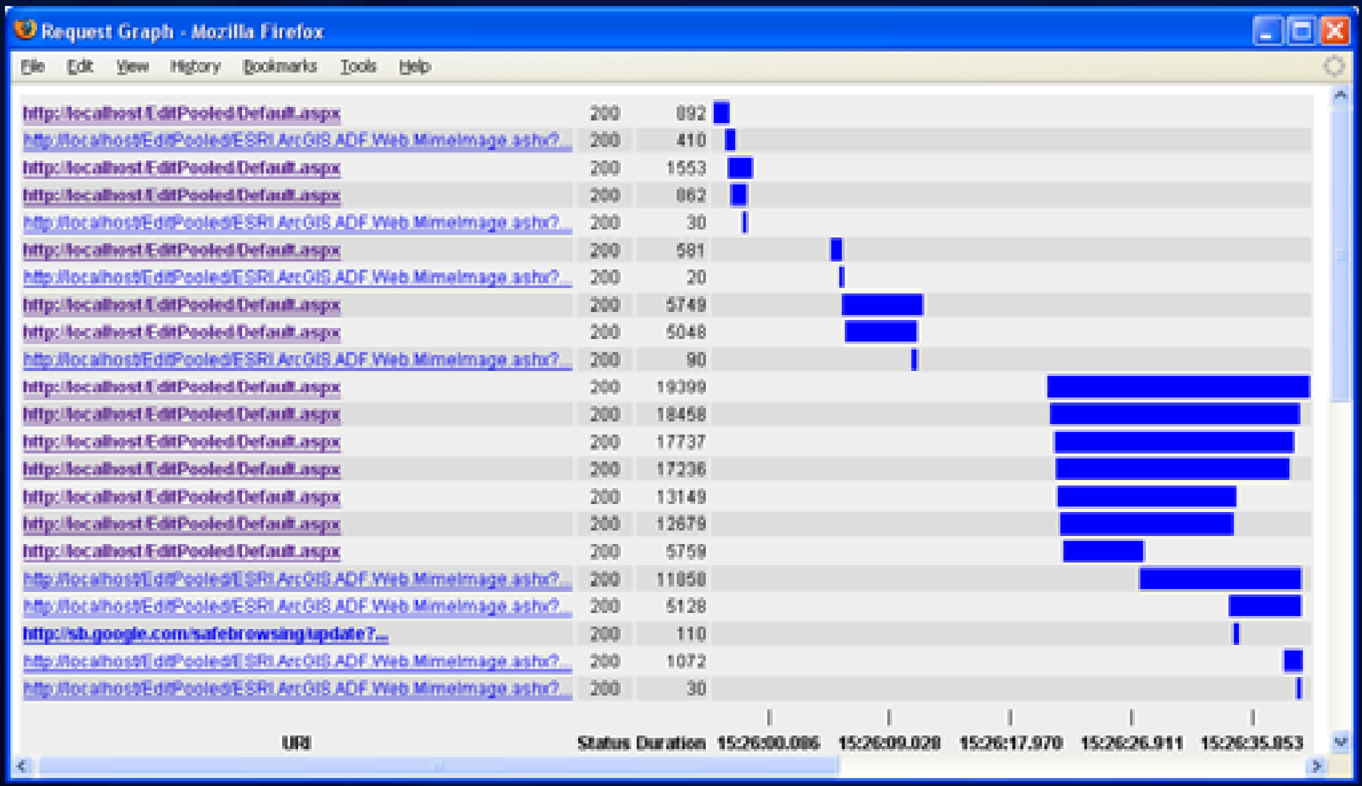


A test is executed at the web browser.

It measures web browser call's elapsed time (roundtrip between browser and data source)

Tuning Primer

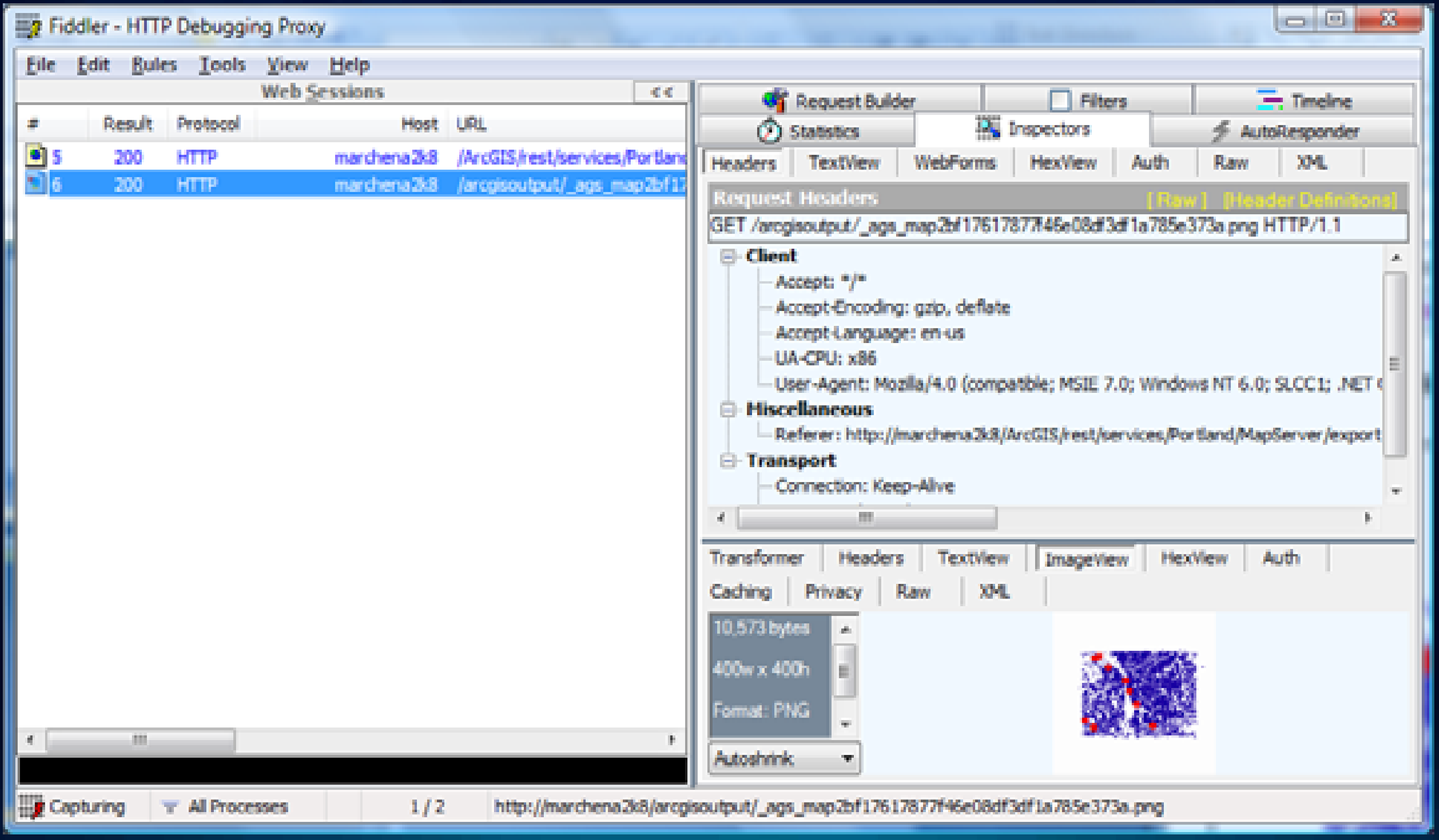
Web diagnostic tools: Fiddler, Tamperdata, Yslow



Tuning Primer

Web diagnostic tools: Fiddler

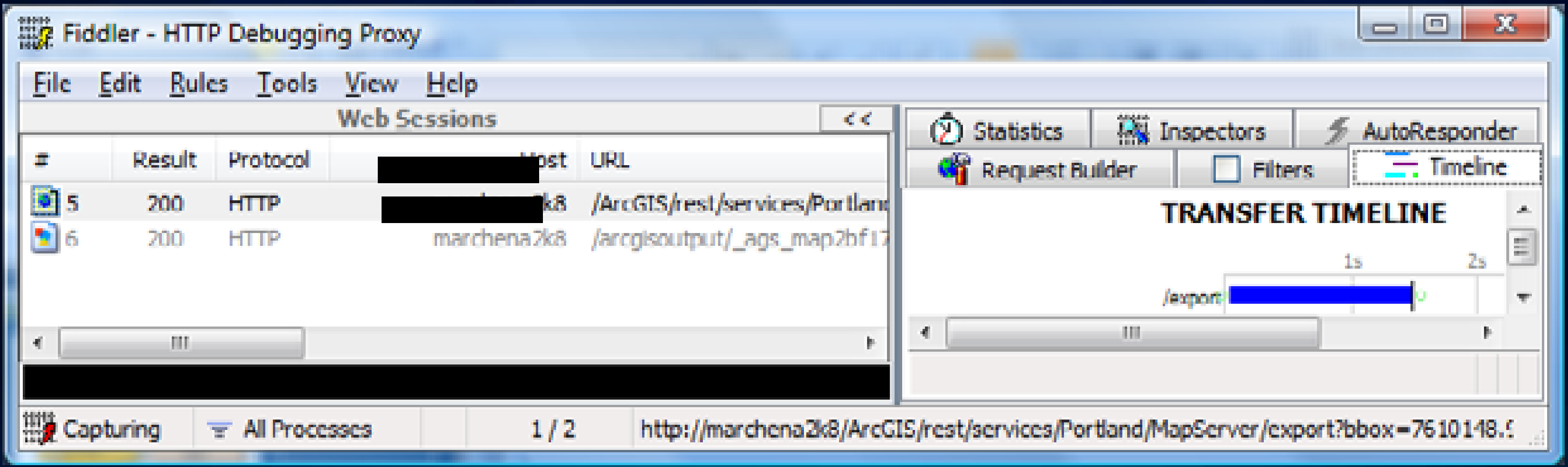
- Can validate image returned



Tuning Primer

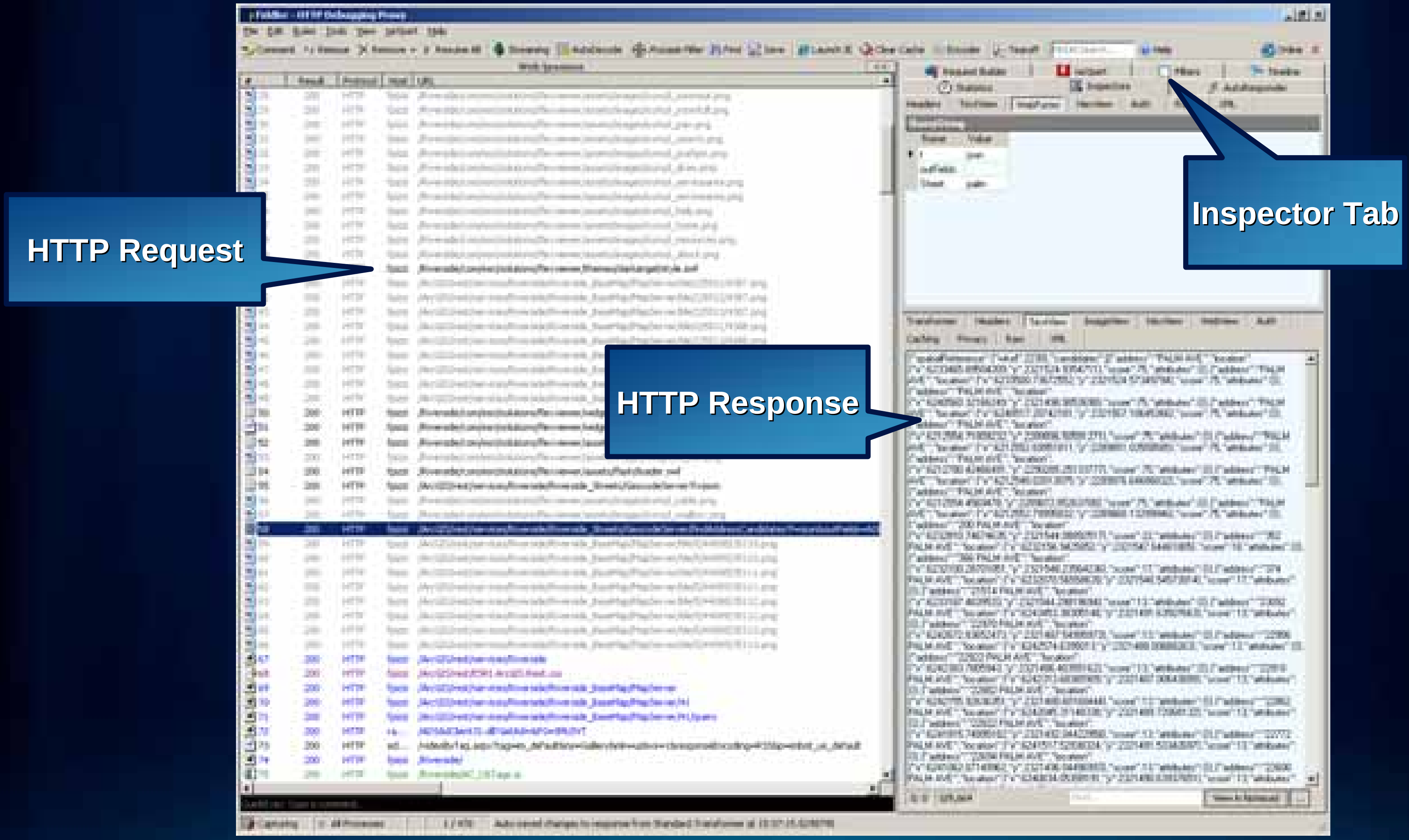
Web diagnostic tools: Fiddler

- **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**



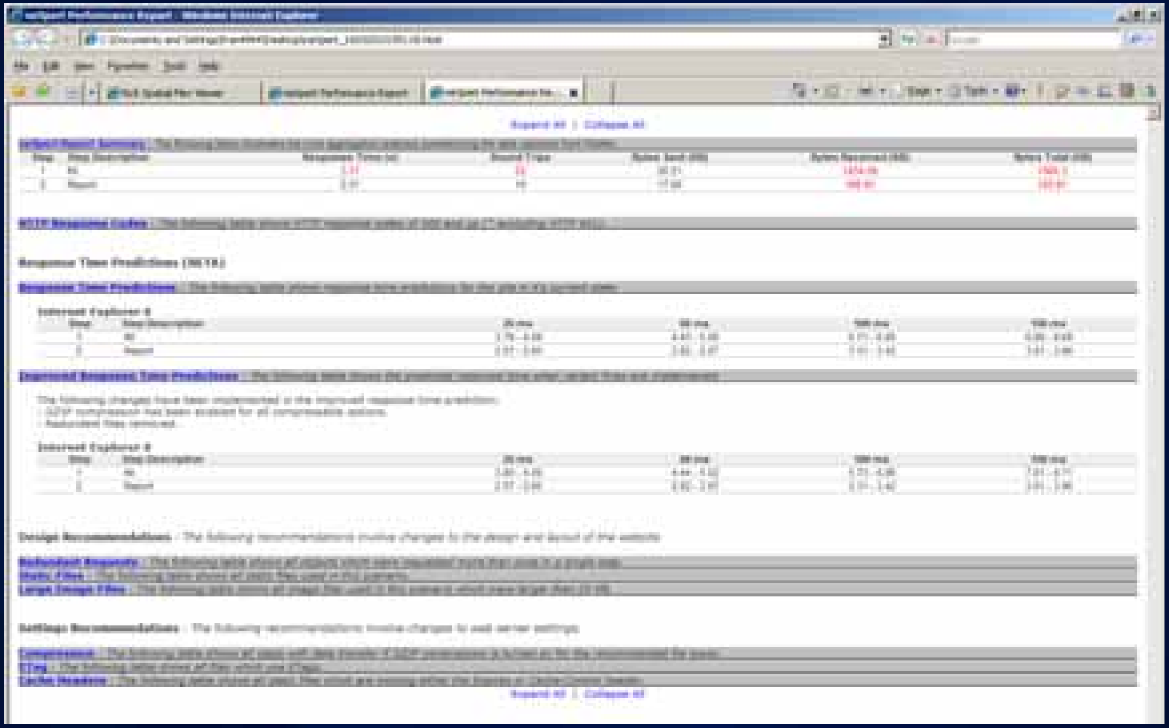
Tuning Primer

Web diagnostic tools: Fiddler



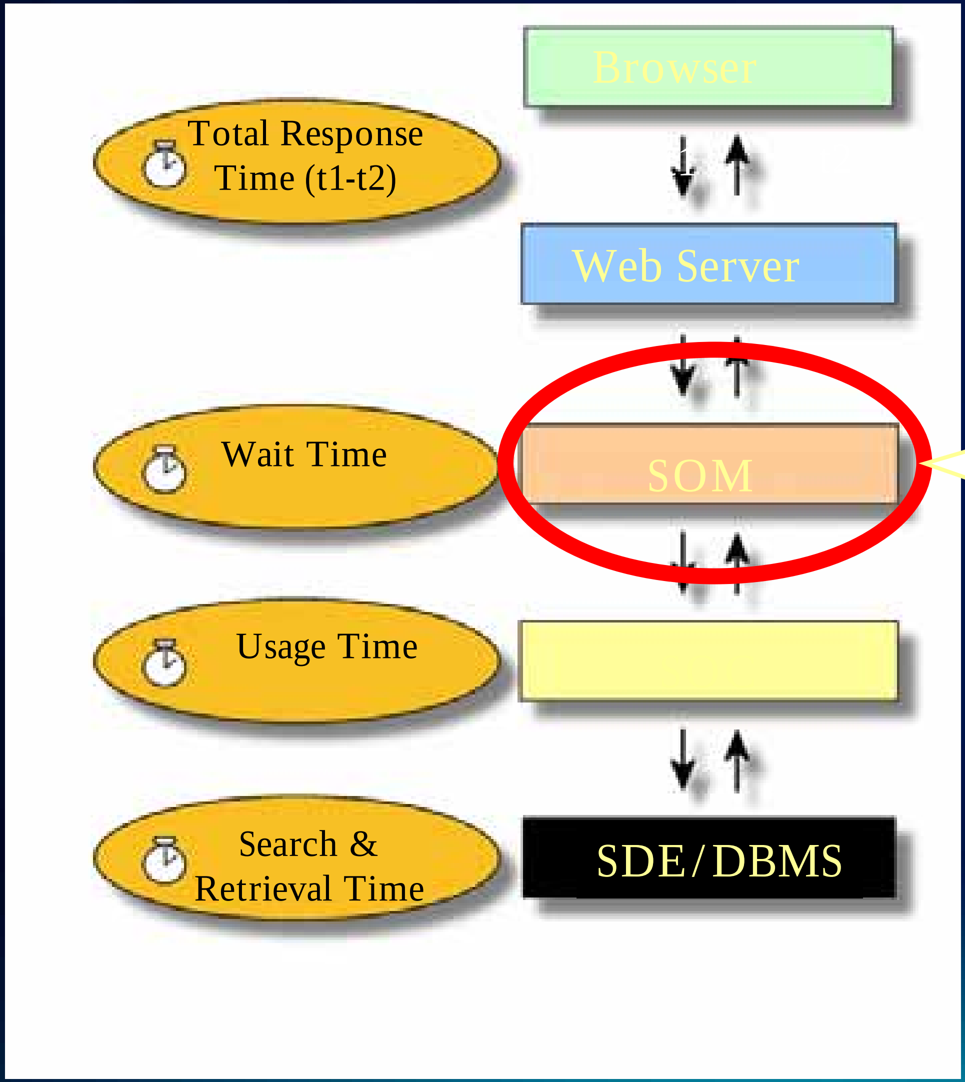
Tuning Primer

Web diagnostic tools: Fiddler (NeXpert Report)



Tuning Primer

Analyze SOM/SOC statistics

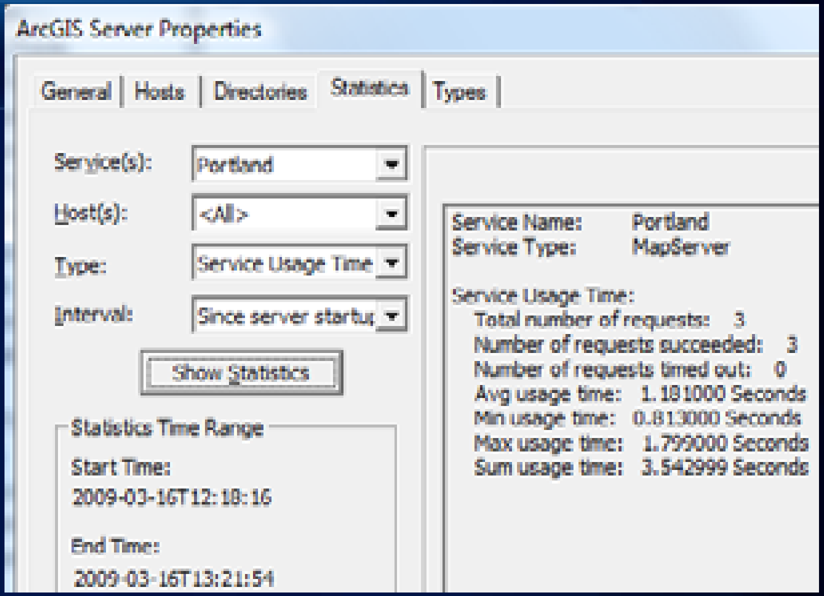


Analyze AGS context server statistics using ArcCatalog, Manager or logs They provide aggregate and detailed information to help reveal the cause of the performance problem.

Tuning Primer

Analyze SOM/SOC statistics

- **ArcCatalog**



- **Detailed log - set to verbose**

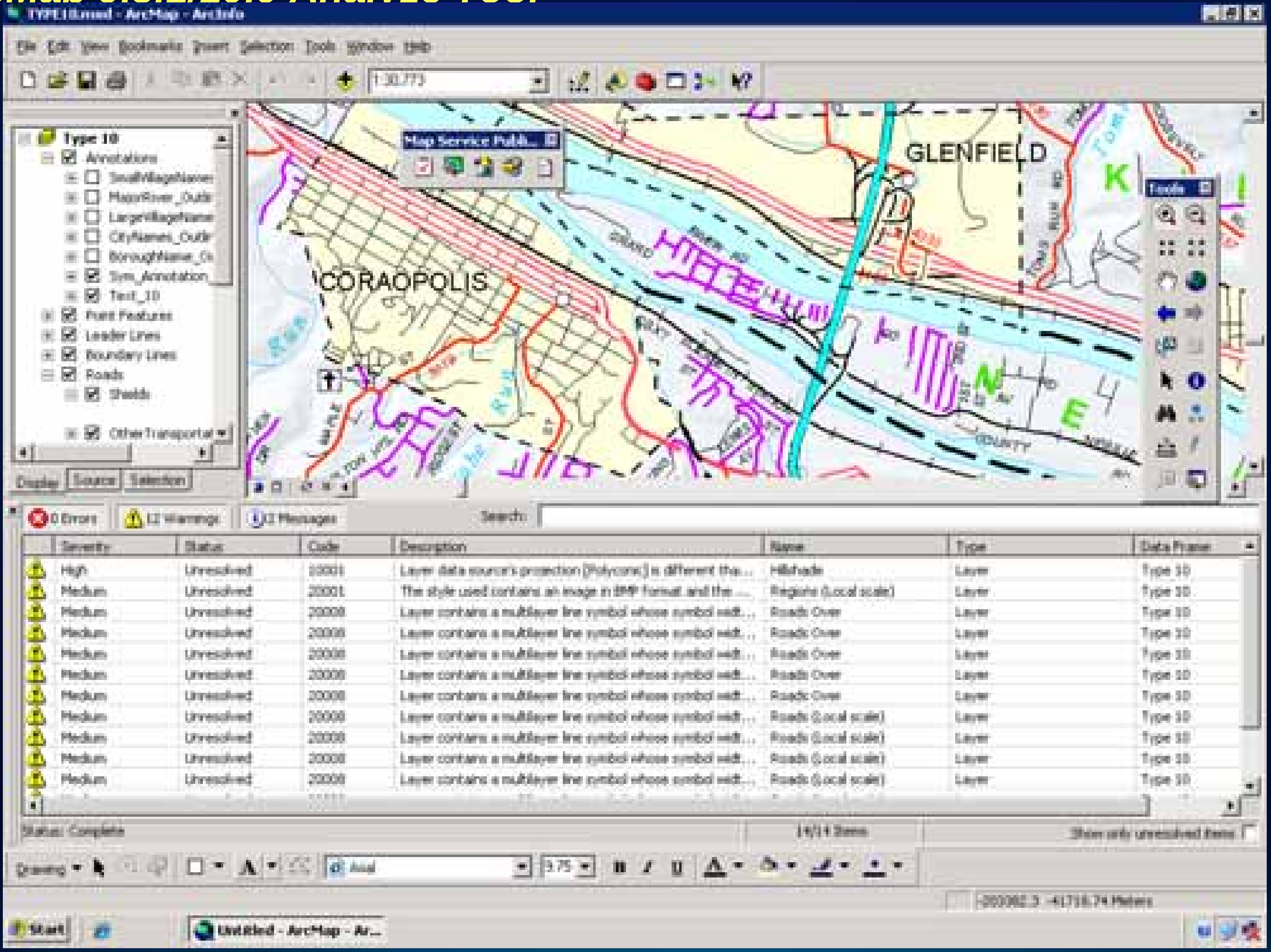
<Msg time="2009-03-16T12:23:22" type="INFO3" code="103021" target="Portland.MapServer" methodName="FeatureLayer.Draw" machine="myWebServer" process="2836" thread="3916" elapsed="0.05221">Executing query.</Msg>

<Msg time="2009-03-16T12:23:23" type="INFO3" code="103019" target="Portland.MapServer" methodName="SimpleRenderer.Draw" machine="myWebServer" process="2836" thread="3916">Feature count: 27590</Msg>

<Msg time="2009-03-16T12:23:23" type="INFO3" code="103001" target="Portland.MapServer" methodName="Map.Draw" machine="myWebServer" process="2836" thread="3916" elapsed="0.67125">End of layer draw: STREETS</Msg>

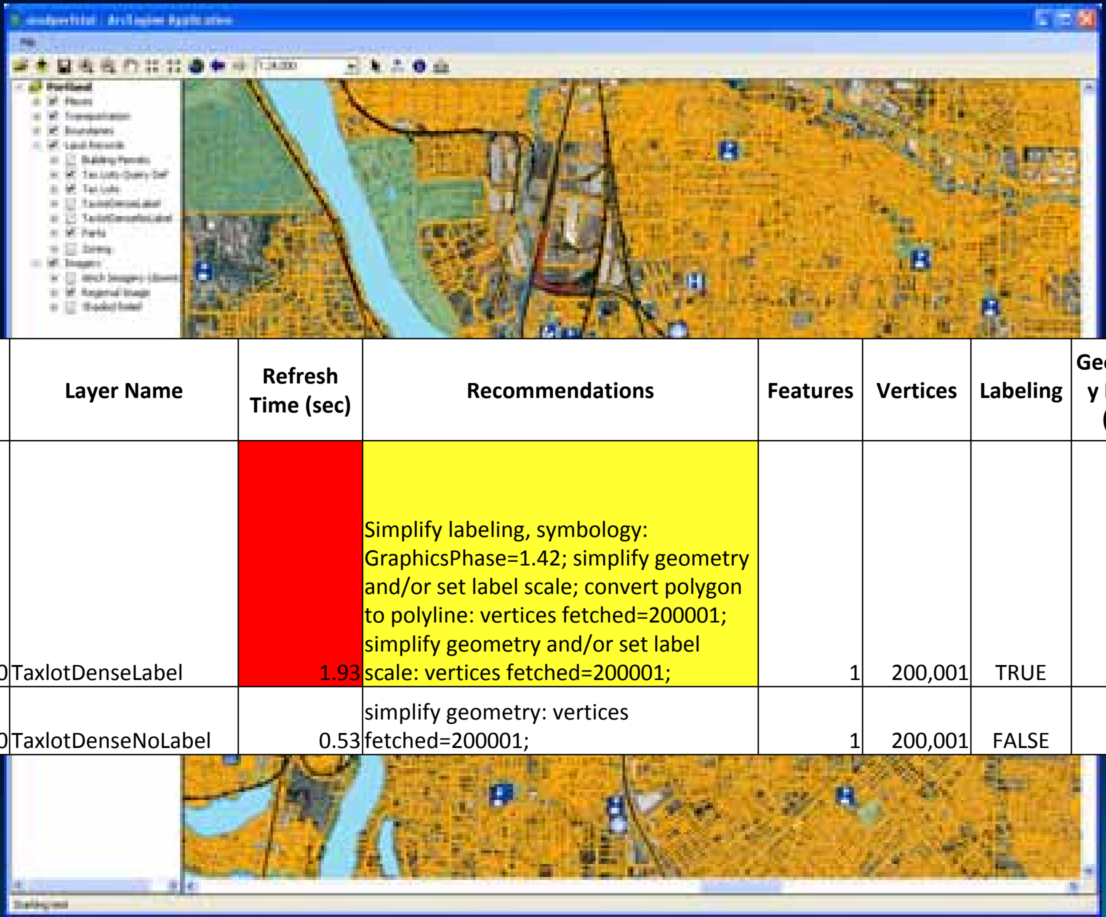
Tuning Primer

ArcMap 9.3.1/10.0 Analyze Tool



Tuning Primer
mxdperfstat

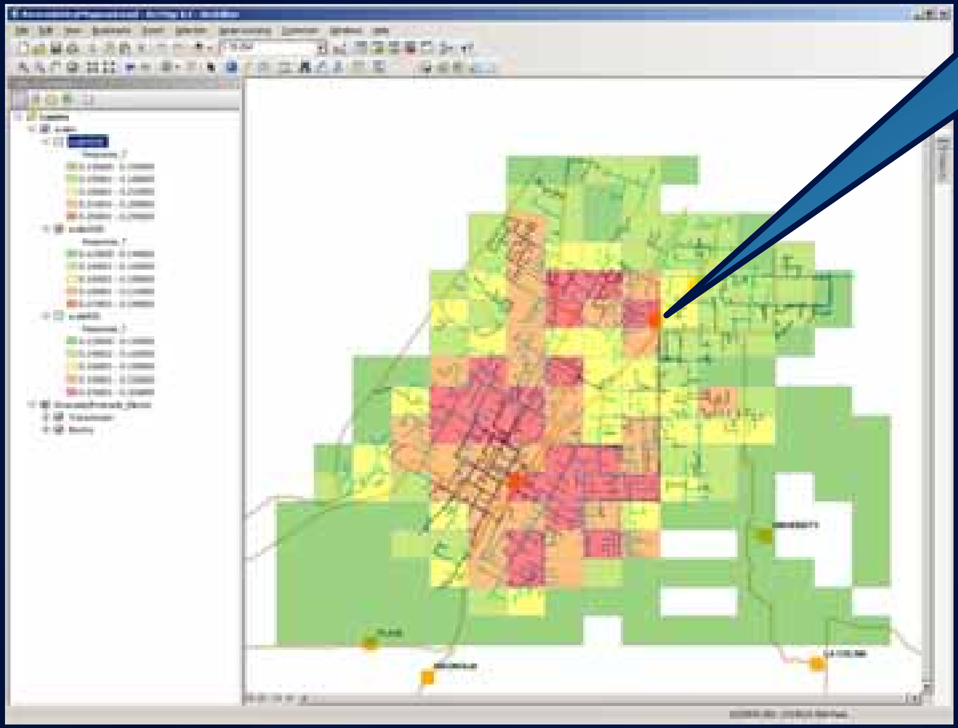
http://resources.arcgis.com/gallery/file/enterprise-gis/details?entryID=6391E988-1422-2418-88DE-3E052E78213C



Item	At Scale	Layer Name	Refresh Time (sec)	Recommendations	Features	Vertices	Labeling	Geography Phase (sec)	Graphics Phase (sec)	Cursor Phase (sec)	DBMS CPU	DBMS LIO
41	1,000	TaxlotDenseLabel	1.93	Simplify labeling, symbology: GraphicsPhase=1.42; simplify geometry and/or set label scale; convert polygon to polyline: vertices fetched=200001; simplify geometry and/or set label scale: vertices fetched=200001;	1	200,001	TRUE	0.45	1.42	1.04	0.02	266
42	1,000	TaxlotDenseNoLabel	0.53	simplify geometry: vertices fetched=200001;	1	200,001	FALSE	0.45	0.02	0.9	0.02	140

Tuning Primer

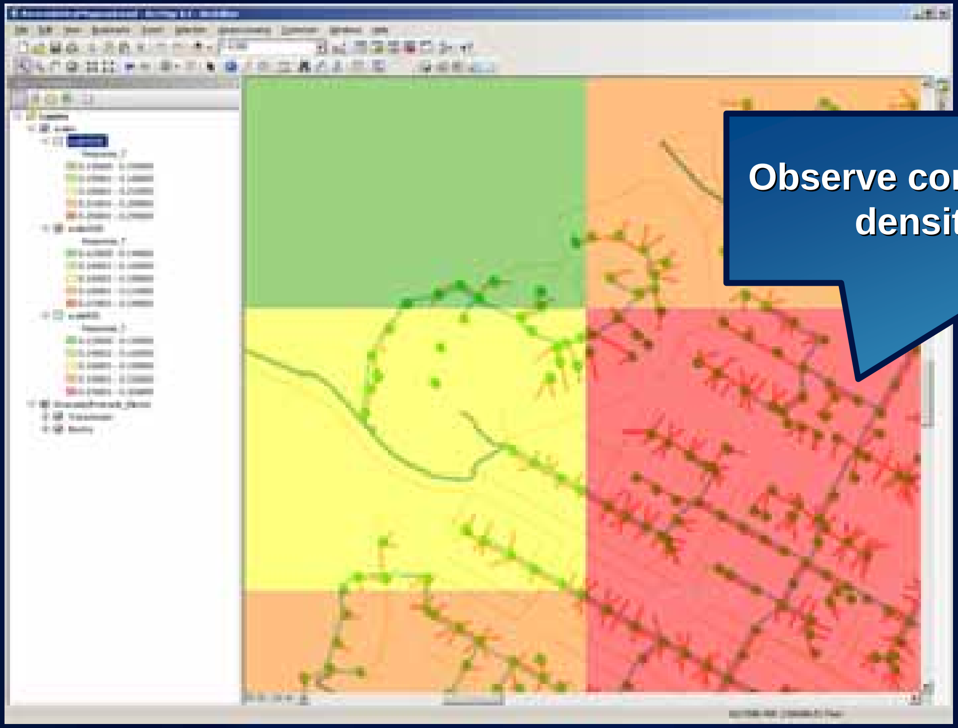
ArcGIS Services



Heat Map based on response times from ArcGIS Server

Tuning Primer

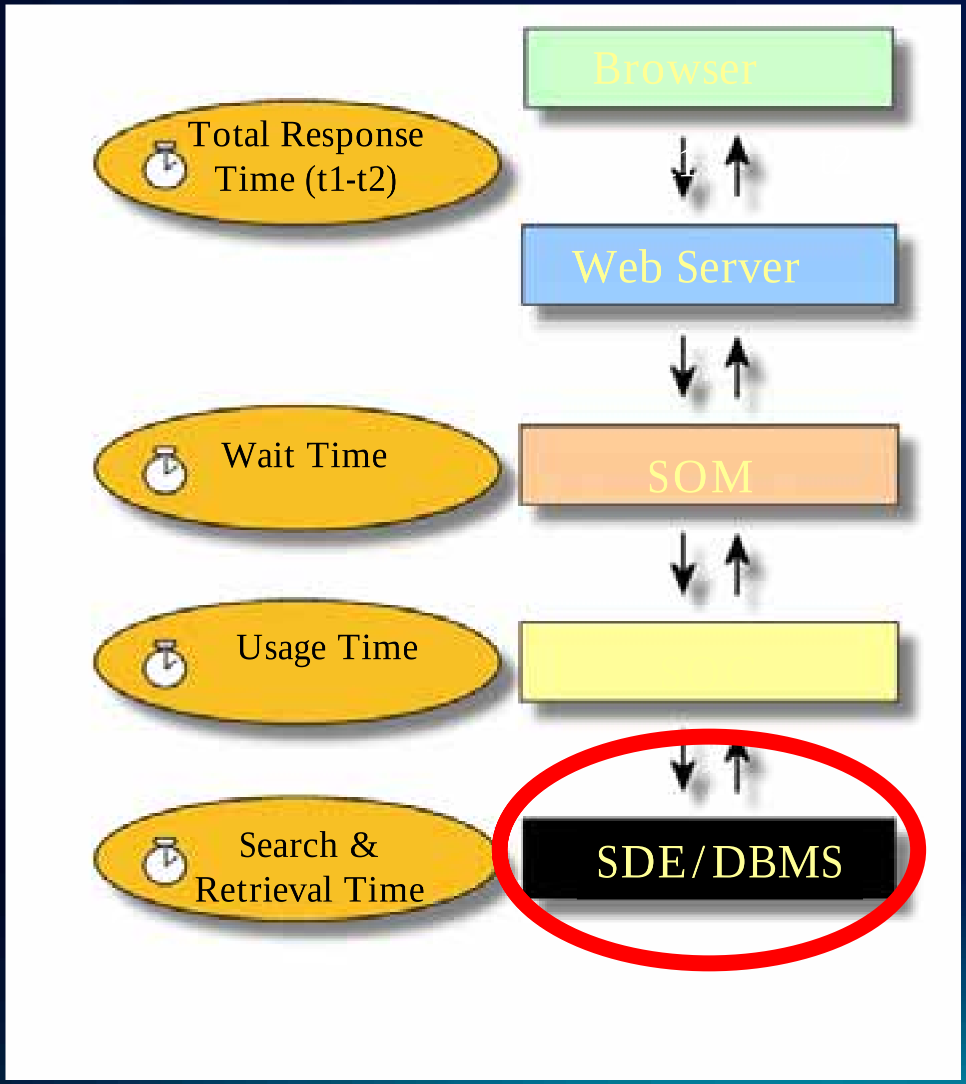
ArcGIS Services



Observe correlation between feature density and performance

Tuning Primer

Data sources



Tuning Primer

Data Sources – Oracle Trace

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

USERNAME     SID     SERIAL# PROGRAM     LOGON_TIM

------------------------------ ---------- ---------- ------------------------------
  -----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

Tuning Primer

Starting Oracle trace using a custom ArcMap UIControl

```
Private Sub OracleTrace_Click()
  .  .    .
  Set pFeatCls = pFeatLyr.FeatureClass
  Set pDS = pFeatCls
  Set pWS = pDS.Workspace
  sTraceName = InputBox("Enter <test_name><email>")
   pWS.ExecuteSQL ("alter session set tracefile_identifier = '" &
  sTraceName & "'")
   pWS.ExecuteSQL ("ALTER SESSION SET events '10046 trace name context
  forever, level 12'")
  .  .    .
End Sub
```

Tuning Primer

Data Sources – Oracle Trace (continued)

```
SQL ID : 71py6481sj3xu
SELECT  1 SHAPE, TAXLOTS.OBJECTID,  TAXLOTS.SHAPE.points,TAXLOTS.SHAPE.numpts,
  TAXLOTS.SHAPE.entity,TAXLOTS.SHAPE.minx,TAXLOTS.SHAPE.miny,
  TAXLOTS.SHAPE.maxx,TAXLOTS.SHAPE.maxy,TAXLOTS.rowid
FROM  SDE.TAXLOTS TAXLOTS WHERE SDE.ST_EnvIntersects(TAXLOTS.SHAPE,:1,:2,:3,:4) =  1
   call     count       cpu    elapsed       disk      query    current        rows
   ------- ------  -------- ---------- ---------- ---------- ----------  ----------
   Parse        0      0.00       0.00          0          0          0           0
   Execute      1      0.07       0.59        115       1734          0           0
   Fetch      242      0.78      12.42       2291      26820          0       24175
   ------- ------  -------- ---------- ---------- ---------- ----------  ----------
   total      243      0.85      13.02       2406      28554          0       24175
                  Elapsed times include waiting on following events:
      Event waited on                             Times   Max. Wait  Total Waited
      ----------------------------------------   Waited  ----------  ------------
      SQL*Net message to client                     242        0.00          0.00
      db file sequential read                      2291        0.39         11.69
      SQL*Net more data to client                   355        0.00          0.02
      SQL*Net message from client                   242        0.03          0.54
   ********************************************************************************
```

Tuning Primer

Data Sources – Oracle Trace (continued)

- Definitions
  - Elapsed time [sec] =(CPU + wait event)
  - CPU [sec]
  - Query (Oracle blocks e.g. 8K read from memory)
  - Disk (Oracle blocks read from disk)
  - Wait event [sec], e.g. db file sequential read
  - Rows fetched

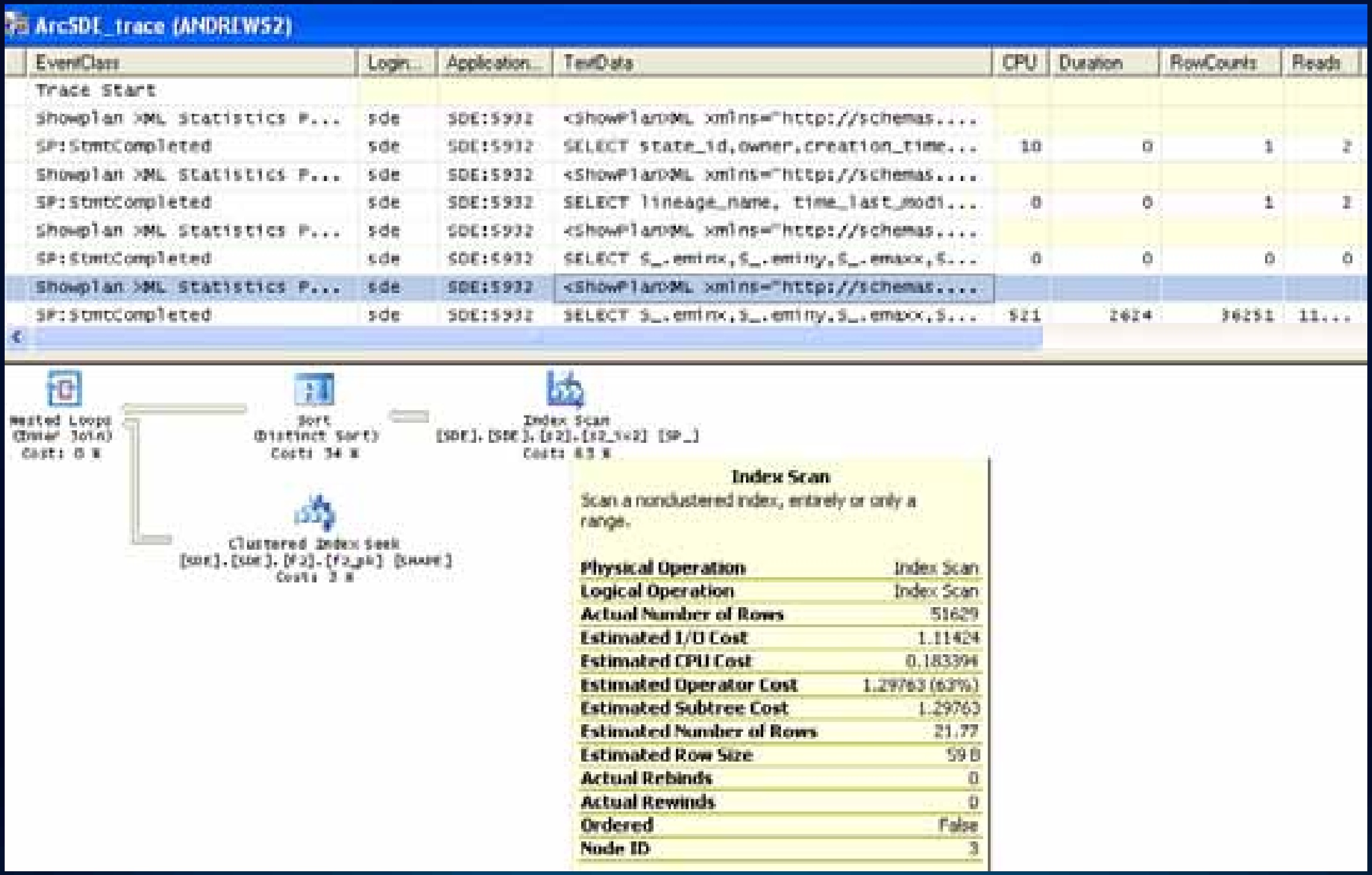
Tuning Primer

Data Sources – Oracle Trace (continued)

- Example (cost of physical reads):
  - Elapsed time = 13.02 sec
  - CPU = 0.85 sec
  - Disk = 2291 blocks
  - Wait event (db file sequential read)=11.69 sec
  - Rows fetched = 24175

Tuning Primer

Data Sources – SQL Profiler



Summary

- Optimize ArcGIS Services
- Profile individual user operations and tune if needed
- Drill down through software stack
  - Application
  - Service
  - Mxd
  - Layer
  - DBMS query
- Correlate your findings between tiers
- Performance and load test

Performance testing

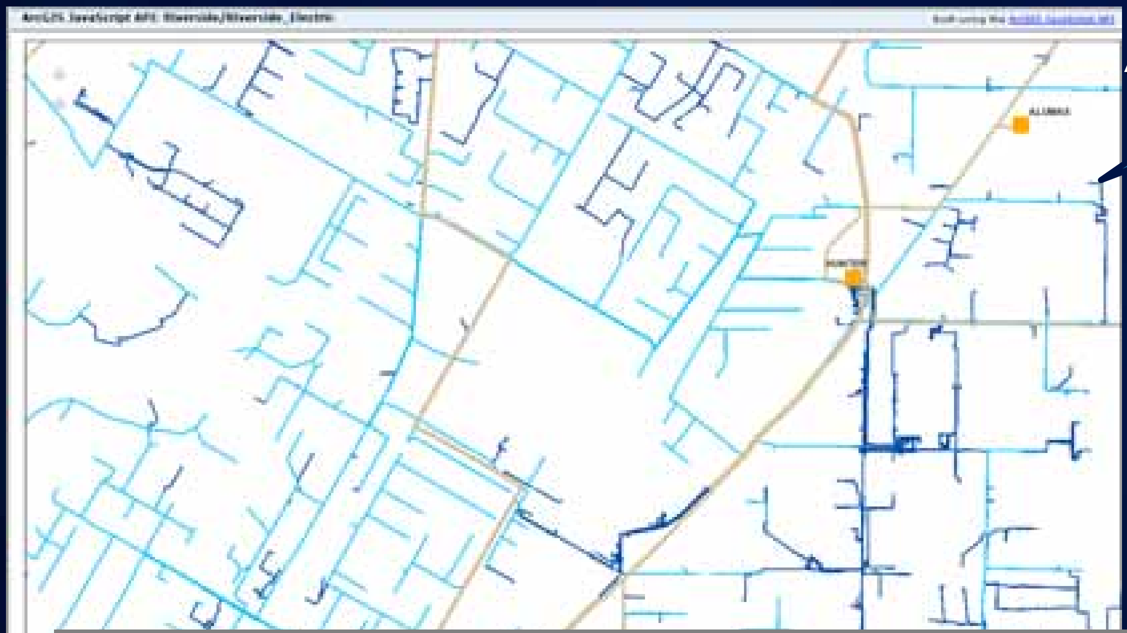
Test Objectives

- Contractual Service Level Agreement
- Bottlenecks
- Capacity
- Benchmark

Test Data

Test Data

Bbox (Using Fiddler)



Area of Interest

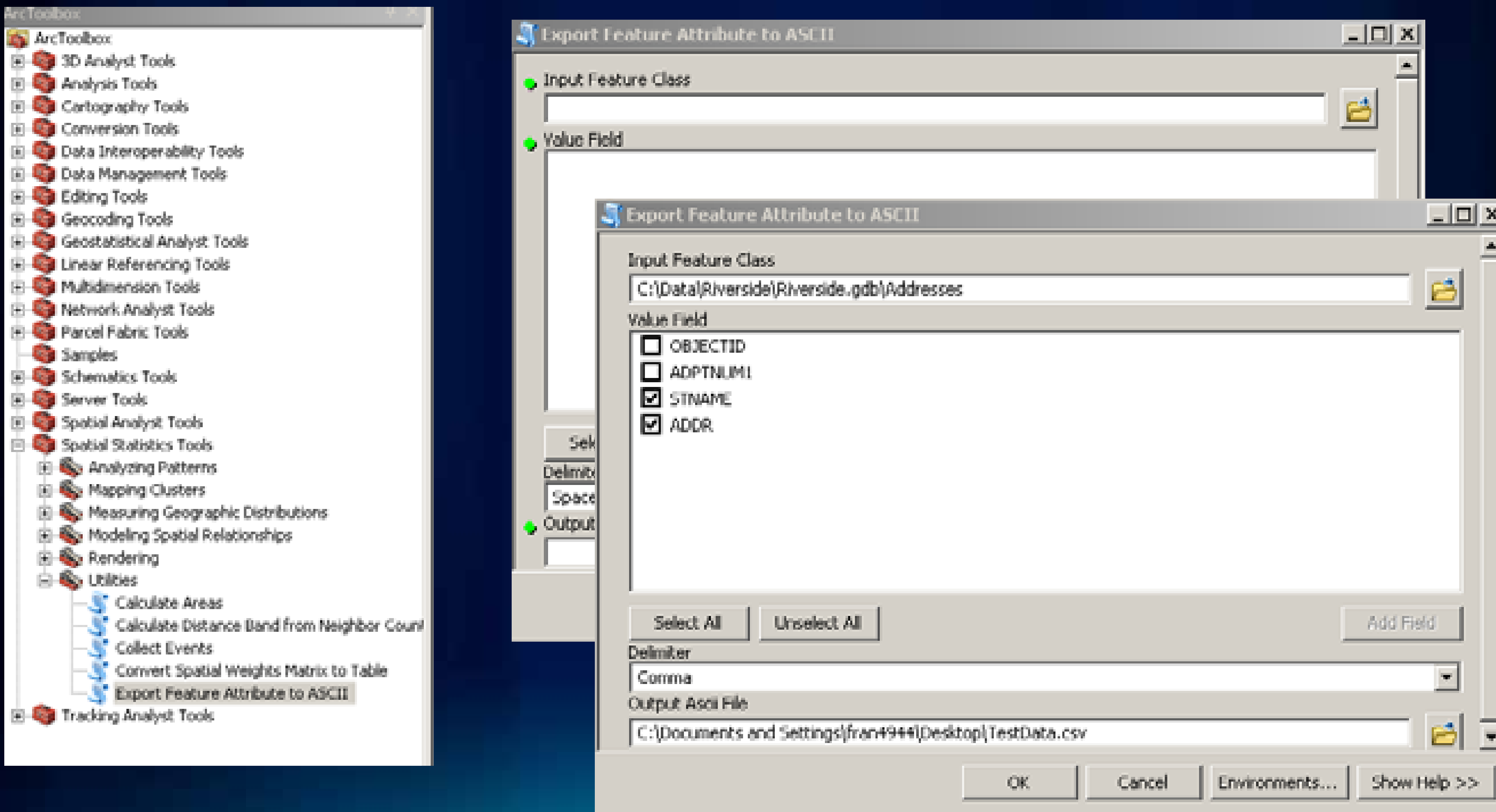
QueryString

Name	Value
f	image
dpi	96
transparent	true
format	png8
bbox	{"xmin":6219593.737018972,"ymin":2303862.765275147,"xmax":6231478.566248391,"ymax":2311468.277924415,"spatialReference":{"wkid":2230}}
bboxSR	2230
imageSR	2230
size	1222,782

Selected Extent
From HTTP Debugging Proxy

Test Data

Attribute Data



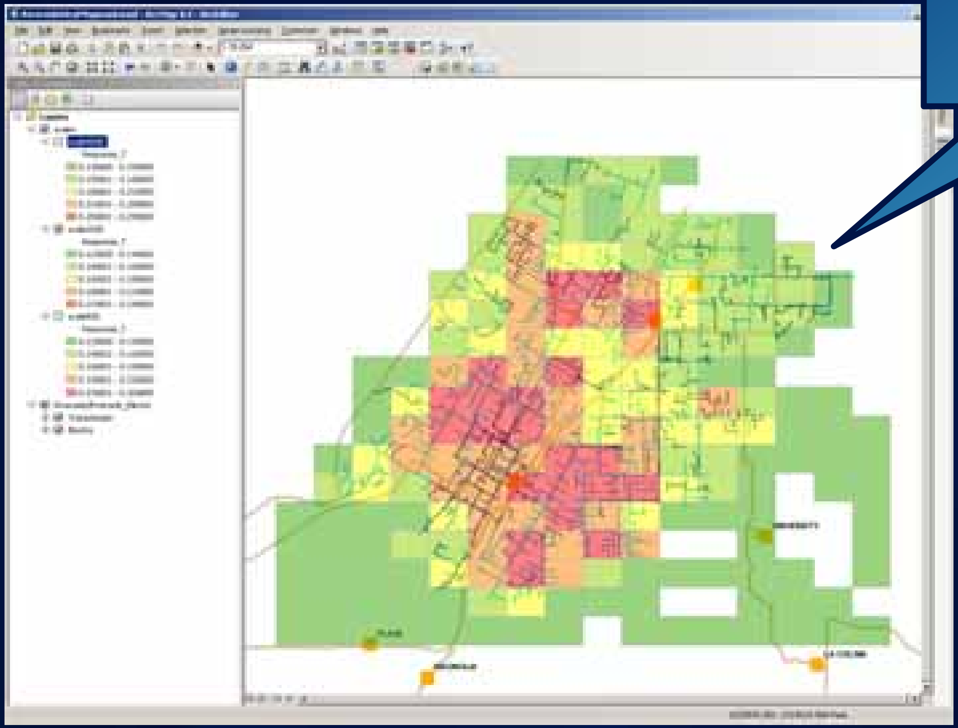
Test Data

Generate Bboxes

One simple example of Python script to generate Bboxes

```python
"""
Generate Bboxes from selected Riverside extent from ...
Note* Image size=1222,702
"""
__author__ = ""
__version__ = "0.1"

import random

def generateBoxs(fullExtent, gridControl):
    bBoxs = []
    bBoxs.append(fullExtent)
    width  = fullExtent[2]-fullExtent[0]
    height = fullExtent[3]-fullExtent[1]
    for grid in gridControl:
        nWidth = width/grid
        nHeight = height/grid
        for row in range(0, grid):
            for column in range(0, grid):
                minX=fullExtent[0]+(column*nWidth)
                minY=fullExtent[1]+(row*nHeight)
                maxX=minX+nWidth
                maxY=minY+nHeight
                bBoxs.append((minX, minY, maxX, maxY))
    return bBoxs

def writeTuple(path, arr):
    try:
        f = open(path, 'w')
        for item in arr:
            f.write(",".join([str(x) for x in item])+"\n")
        f.close()
    except IOError, (errno, strerror):
        print path
        print "writeTuple I/O error(%s): %s" % (errno, strerror)
    except:
        print "writeTuple Unexpected error:", sys.exc_info()[0]
        raise

if __name__=="__main__":

    extent =(6219593.737018972,2303862.765275147,6231478.566248391,2311468.277924415)
    grid = [2, 8, 16]
    bBoxs = generateBoxs(extent, grid)
    for item in bBoxs:
        print item
    writeTuple("C:\\test.csv", bBoxs)
```

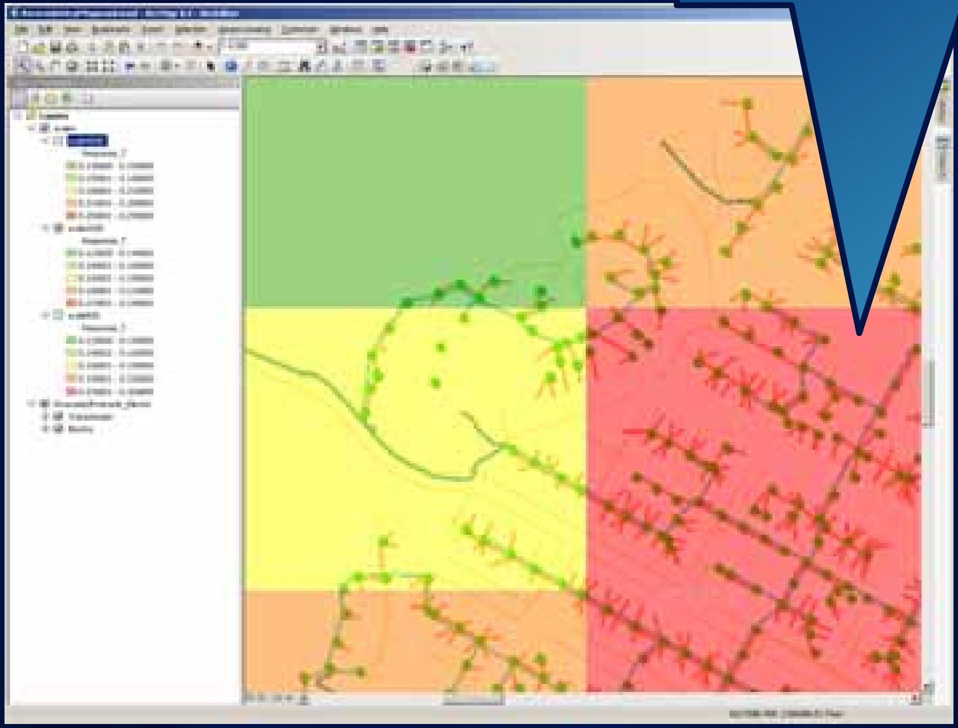
Test Data



Heat Map based on response times from ArcGIS Server

Test Data

Observe correlation between feature density and performance

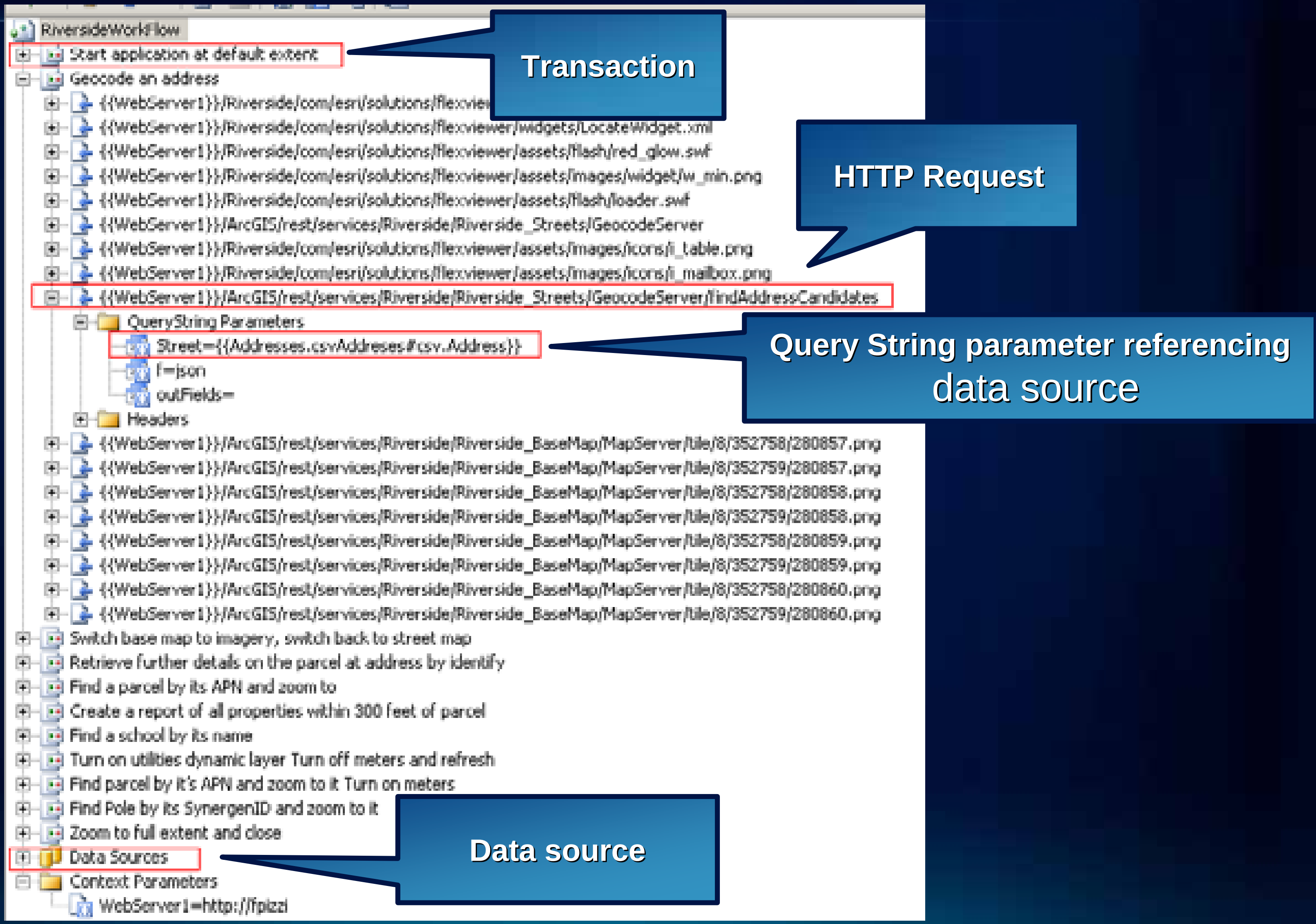


TEST SCRIPTS

Test Scripts

- Record user workflow based on application user requirements
- Create single user web test
  - Define transactions
  - Set think time and pacing based on application user requirements
  - Parameterize transaction inputs
  - Verify test script with single user

Visual Studio Quick Introduction - WebTest

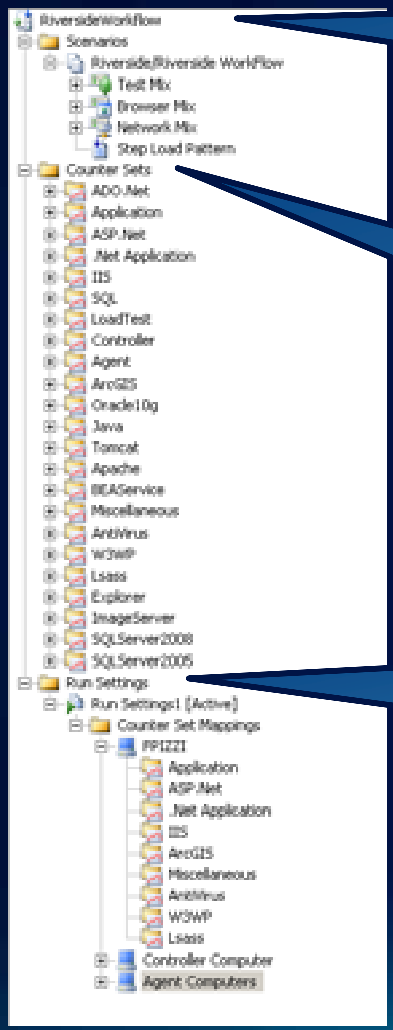


LOAD TEST

Load Test

- Create load test
  - Define user load
  - Create machine counters to gather raw data for analysis
- Execute

Visual Studio Quick Introduction – Load Test



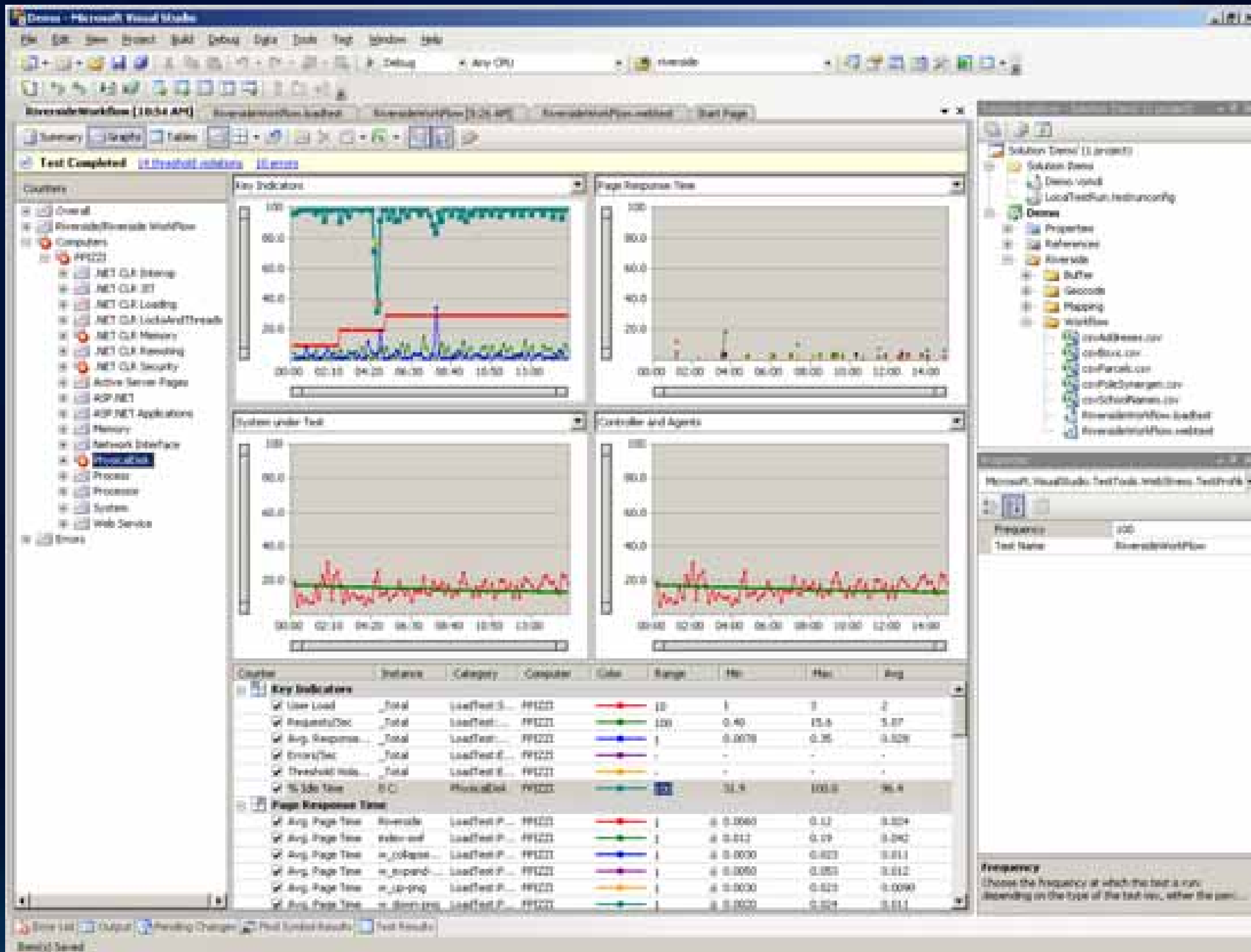
Scenarios:
Test Mix (WebTest or Unit Test),
Browser Mix,
Network Mix,
Step Loads

Perfmon Counter Sets:
Available categories that may be mapped to a machine in the deployment

Run Settings:
Counter Set Mappings – Machine metrics
Test duration

Visual Studio - Load Test Run

Threshold rules violated



Testing with GIS Test Tool

GIS Test Tool

GIS Test Tool– Performance Test Capabilities

- Define web tests including QA step for verification
- Define transactions including think time
- Define load tests
- Execute load tests
- Capture system metrics for multiple machines
- View results and export to Excel

ANALYSIS

Tips and Tricks

Analyze Results - Validation

- Compare and correlate key measurements
  - Response Time (increasing, higher than initially profiled for single user)
  - Throughput
  - CPU on all tiers
  - Network on all tiers
  - Disk on all tiers
  - Passed tests
  - Failed test

Tips and Tricks

Analyze Results - Validation

- Lack of errors does not validate a test
  - Requests may succeed but return zero size image
  - Spot check request response content size

Tips and Tricks

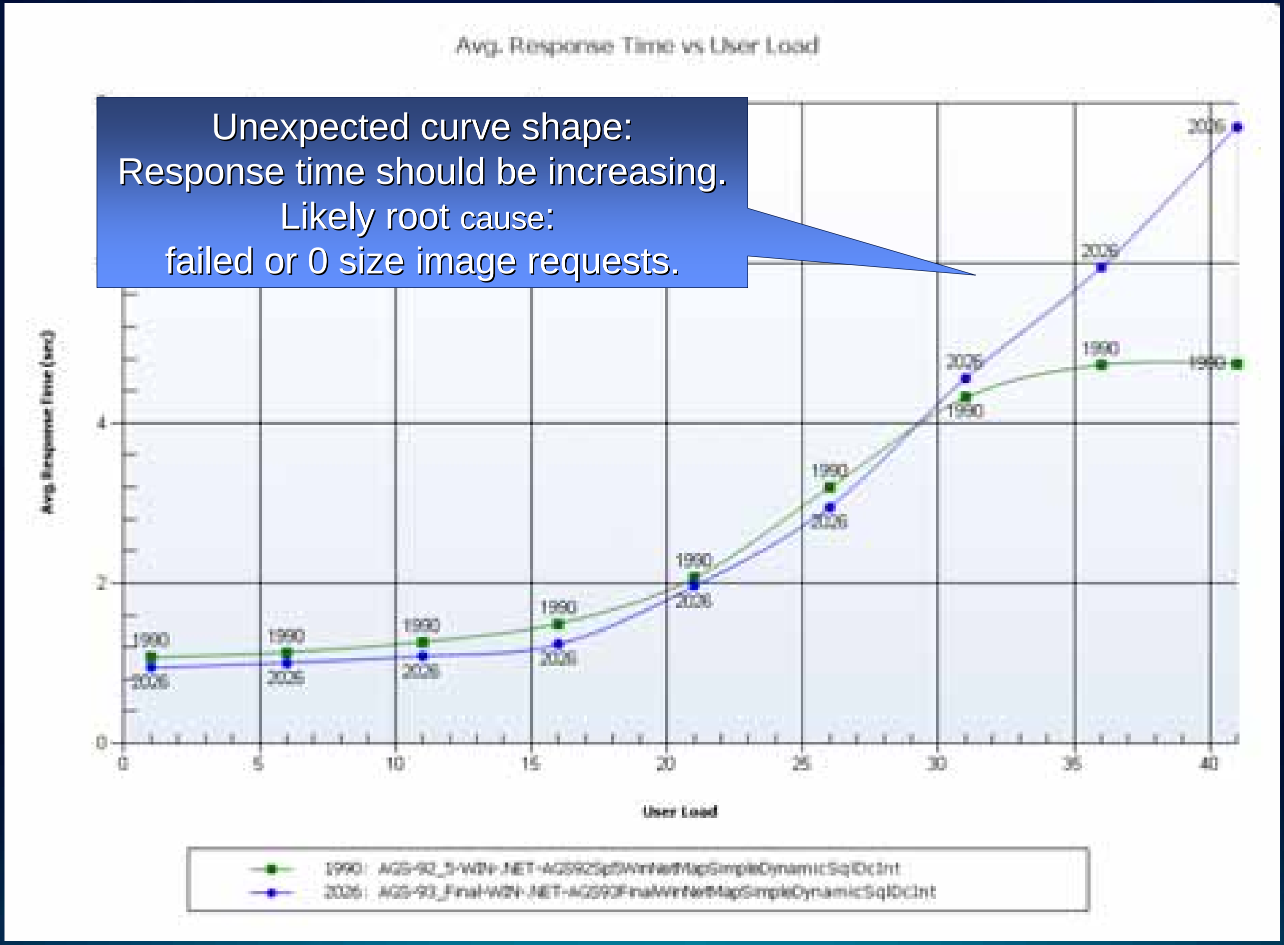
Analyze Results – Reporting and Analysis

- Exclude failure range, e.g. failure rate > 5% from the analysis
- Exclude excessive resource utilization range

Tips and Tricks

Analyze Results

Validation Example – Unexpected response time decrease under heavy load

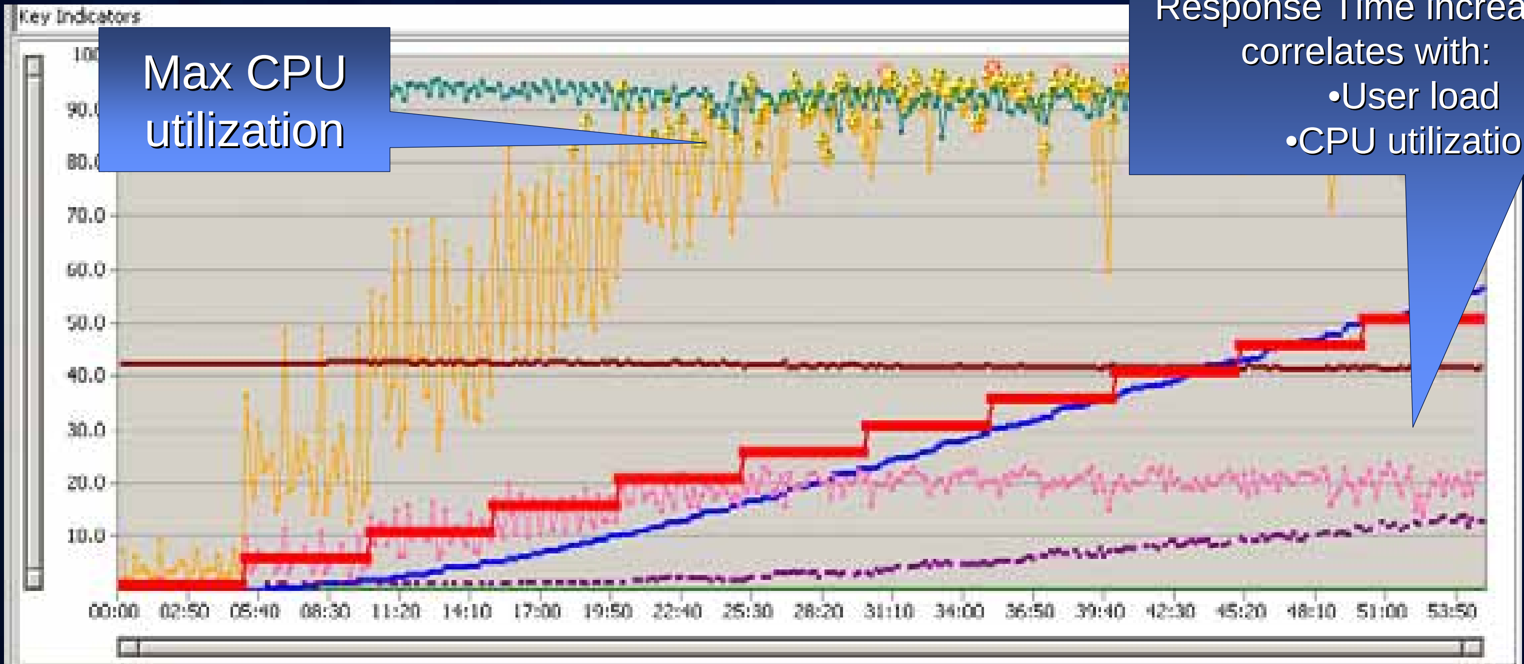


Tips and Tricks

Validation Example – Expected CPU and Response Time Correlation

Max CPU utilization

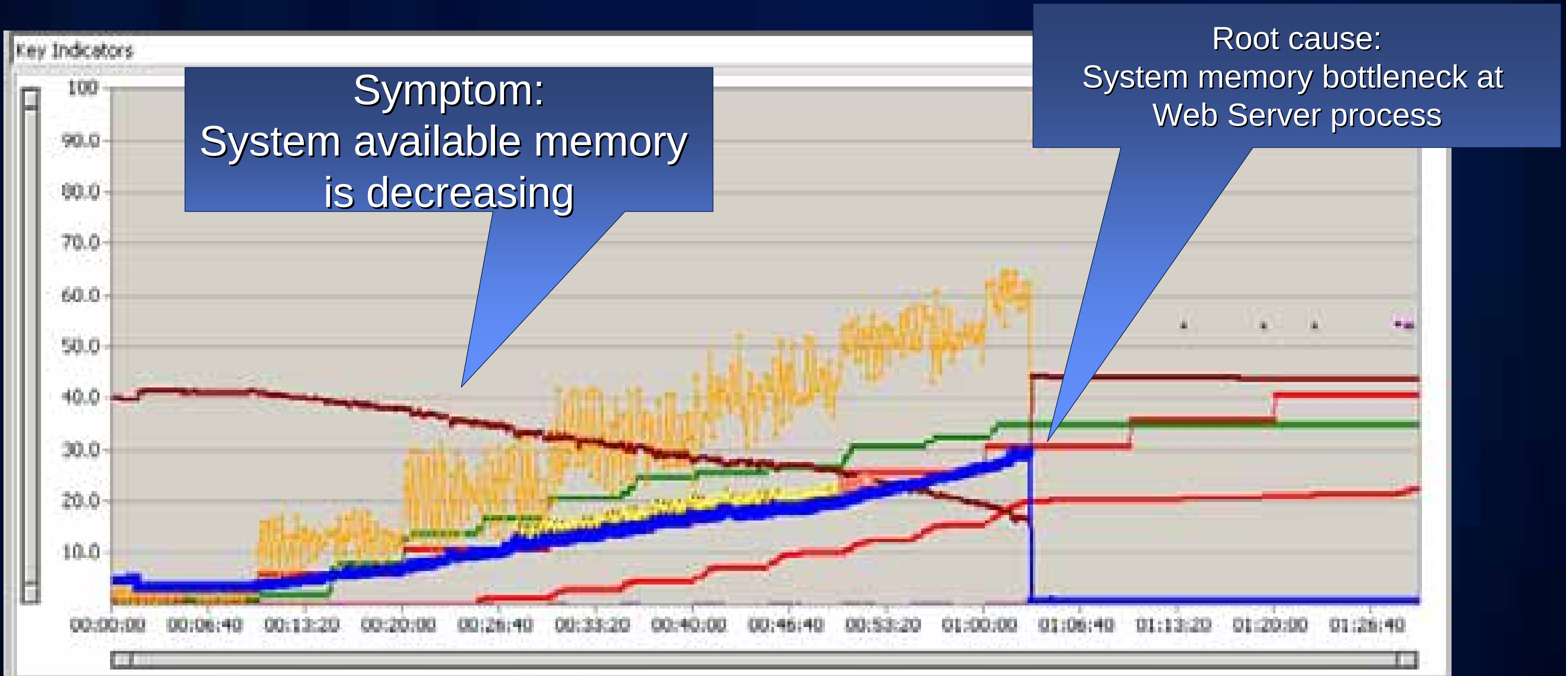
Response Time increase correlates with:
- User load
- CPU utilization



Counter	Instance	Category	Computer	Color	Range	Min	Max	Avg
User Load	_Total	LoadTest:Scenar	ESLW59		100	1	51	26
Failed Tests	_Total	LoadTest:Test	ESLW59		0	0	0	0
Passed Tests	_Total	LoadTest:Test	ESLW59		1,000	0	568	223
Avg. Response Time	_Total	LoadTest:Transa	ESLW59		100	1.27	14.2	6.13
% Processor Time	_Total	Processor	ESLSRV9		100	2.27	96.2	68.4
% Idle Time	_Total	PhysicalDisk	ESLSRV9		100	82.6	98.9	93.0
Bytes Received/sec	Broadcom BCM57	Network Interfac	ESLSRV9		1,000,000	6,632	238,324	158,332
Available MBytes	-	Memory	ESLSRV9		10,000	4,172	4,320	4,242

Tips and Tricks

Validation Example – Test failure due to w3wp memory bottleneck



Counter	Instance	Category	Computer	Color	Range	Min	Max	Avg
☑ User Load	_Total	LoadTest:Scenar	ESLIBMBL4		100	1	41	21
☑ Failed Tests	_Total	LoadTest:Test	ESLIBMBL4		1,000	0	226	99
☑ Passed Tests	_Total	LoadTest:Test	ESLIBMBL4		100	0	35	23
☑ Avg. Response Time	_Total	LoadTest:Transa	ESLIBMBL4		1,000	0.31	546	1.84
☑ % Processor Time	_Total	Processor	ESLSRV15		100	0.94	100	49.6
☑ Available MBytes	-	Memory	ESLSRV15		10,000	1,629	4,485	3,575
☐ Avg. Content Length	ESRI-ArcGIS-ADI	LoadTest:Reque	ESLIBMBL4		1,000,000	5,442	105,690	98,554
☑ Private Bytes	w3wp	Process	ESLSRV15		10,000,000...	46,809,088	3,000,823,808	951,365,184

Tips and Tricks

Determining System Capacity

- Maximum number of concurrent users corresponding to, e.g.:
  - Maximum acceptable response time
  - First failure or 5%
  - Resource utilization greater than 85%, for example CPU
- Different ways of defining acceptance criteria (performance level of service), e.g.
  - 95% of requests under 3 sec
  - Max request under 10 sec

Tips and Tricks - Execute

- Ensure
  - 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

REPORT

Report

- Executive Summary
- Test Plan
  - Workflows
  - Work load
- Deployment documentation
- Results and Charts
  - Key Indicators, e.g. Response Time, Throughput
  - System Metrics, e.g. CPU %
  - Errors
- Summary and Conclusions
  - Provide management recommendations for improvements
- Appendix

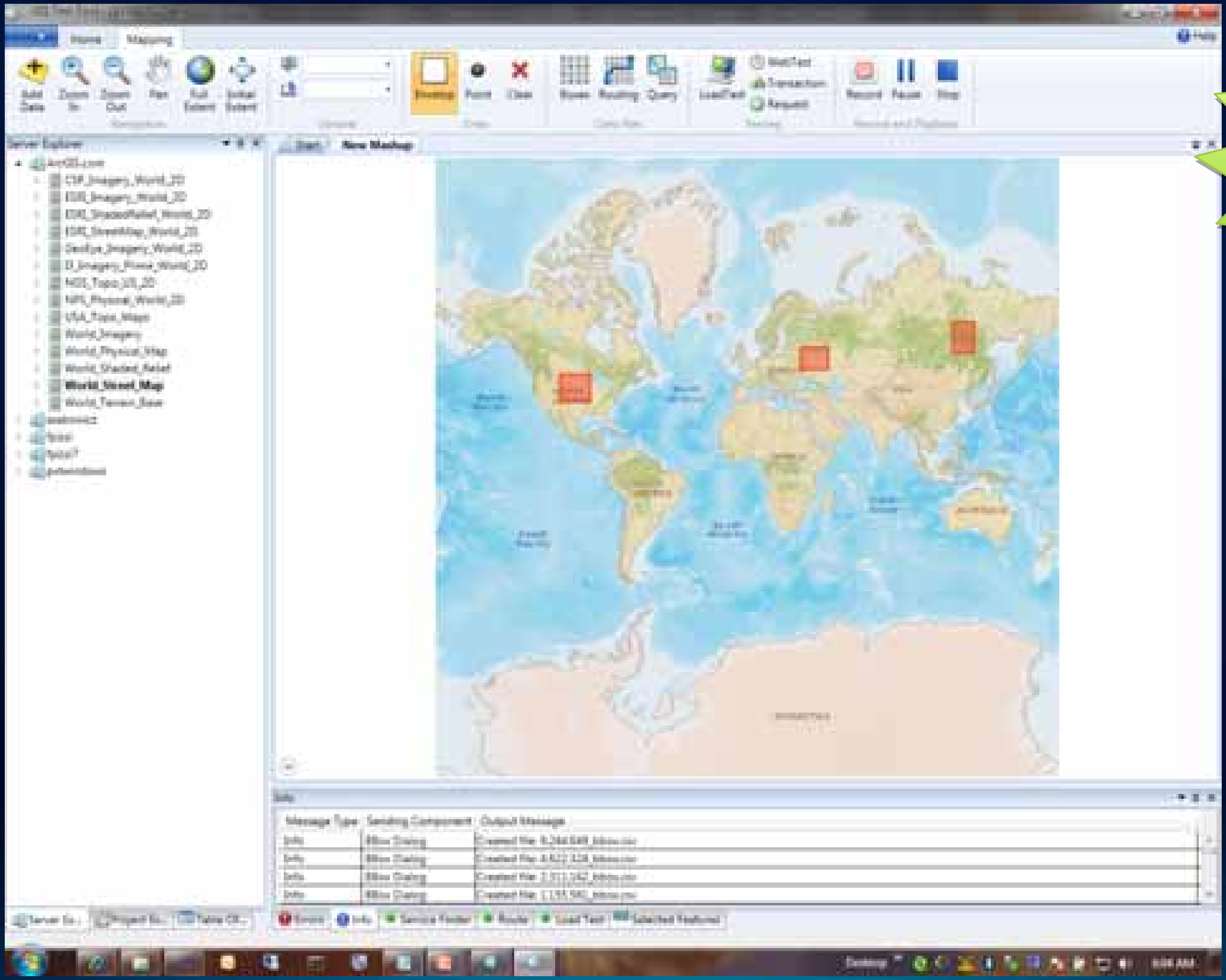
TEST TOOLS

Testing - Selecting Load Test Tool

Test Tools	Open Source	Pros	Cons
LoadRunner	No	•Industry Leader •Automatic negative correlations identified with service level agreements •Http Web Testing •Click and Script •Very good tools for testing SOA • Test results stored in database •Thick Client Testing •Can be used for bottleneck analysis	•High Cost •Test Development in in C programming language •Test metrics difficult to manage and correlate •Poor user community with few available examples
Silk Performer	No	•Good solution for testing Citrix •Wizard driven interface guides the user •Can be used for bottleneck analysis	•Moderate to High Cost •Test metrics are poor •Test Development uses proprietary language •Test metrics difficult to manage and correlate •Poor user community with few available examples
Visual Studio Test Team	No	•Low to moderate cost •Excellent Test Metric reporting •Test Scripting in C# or VB .NET •Unit and Web Testing available •Blog support with good examples •Very good for bottleneck analysis	•No built in support for AMF •No Thick Client options •Moderate user community
JMeter	Yes	•Free •Tool	•Provides only response times •Poor User community with few available examples

GIS Test tool

Quick Preview



Testing - Selecting Load Test Tool

- 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

CAPACITY PLANNING

Input Capacity Planning

- **Find Input for Capacity Planning**
  - **Test Report**
    - **Includes Throughput (Transactions per hour)**
    - **Includes System Metrics – %CPU Utilization, #Cores**
    - **Spec Rate from the machines tested**
  - **Use this information to calculate <u>Service Time</u> for Transactions**

Input Capacity Planning

- **Capacity model expressed as Service Time**

$$ST = \frac{\#CPU \times 3600 \times \%CPU}{TH \times 100}$$

Capacity Planning

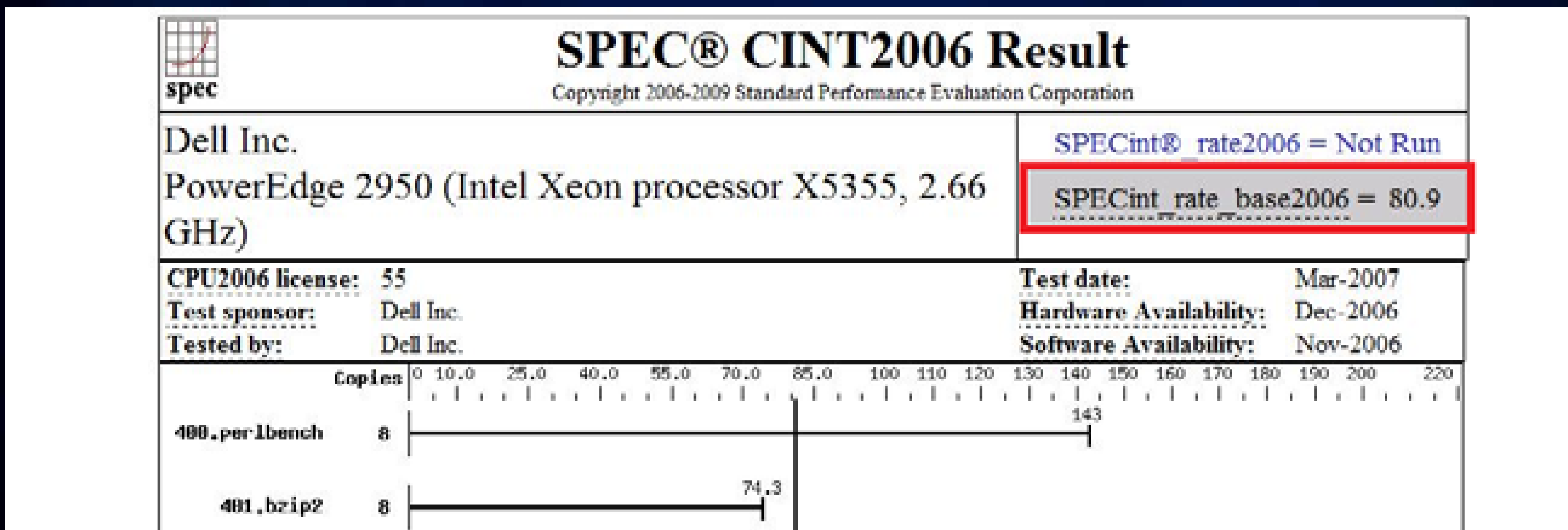
- **Estimate capacity for a different hardware platforms**
  - **Find your target (t) server on http://www.spec.org/cpu2006/results/rint2006.html**

$$\#CPU_t = \frac{ST_b \times TH_t \times 100}{3600 \times \%CPU_t} \times \frac{SpecRatePerCPU_b}{SpecRatePerCPU_t}$$

Capacity Planning

- SPEC.org
  - http://www.spec.org/cpu2006/results/rint2006.html
  - Find your server from the results

Capacity Planning



Capacity Planning

- Additional examples on capacity planning can be found here:
  - http://resources.arcgis.com/gallery/file/enterprise-gis/details?entryID=6367F821-1422-2418-886F-FCC43C8C8E22