Esri Federal User Conference

GIS: The Geographic Approach for the Nation



ArcGIS Server Performance and Scalability-Optimization and Testing

Andrew Sakowicz, Esri Professional Services

Objective

- Overview:
  - Key performance factors
  - Optimization techniques
  - Testing Methodology

Audience

- Audience
  - Testers
  - Developers
  - System Architects
- Level:
  - Intermediate

Performance Factors in ArcGIS Server Components

- ArcGIS Server services
- Data Sources
- ArcGIS Server Framework
- Client applications (Flex, Silverlight, JS, ADF)
- Hardware

Goal: Deliver highly performing deployments.

Performance Factors: ArcGIS Server Services

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

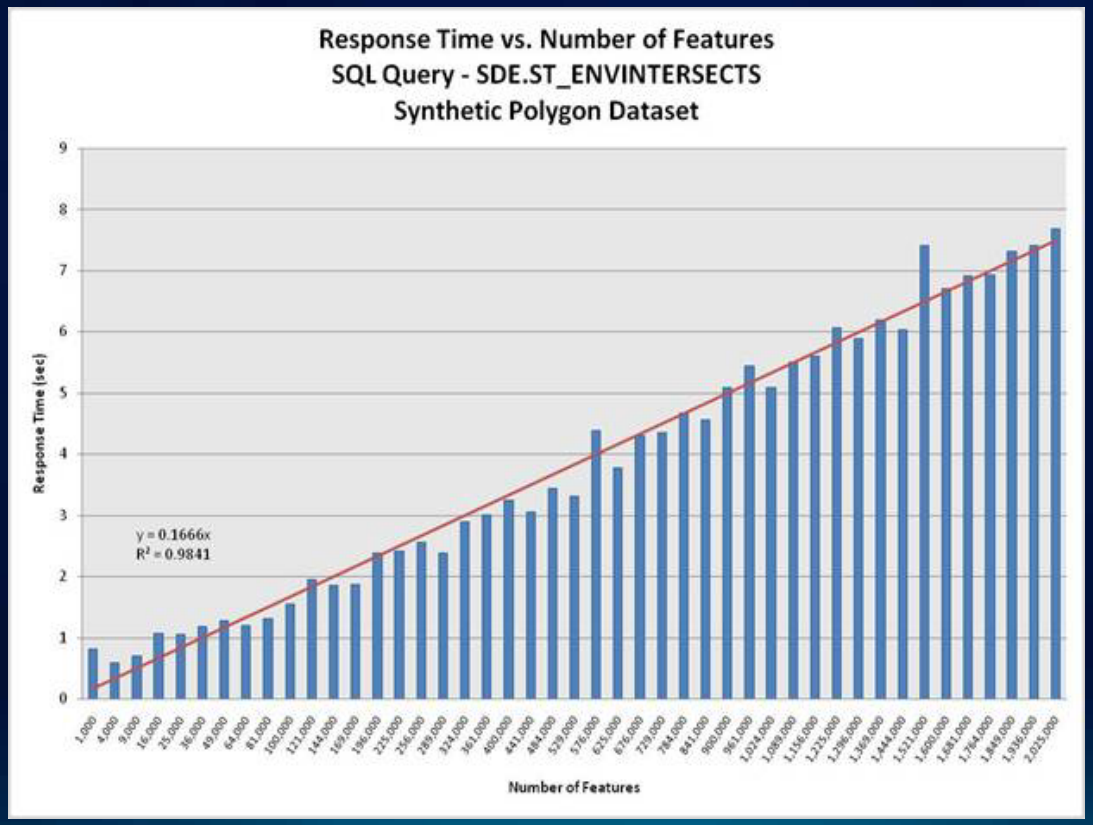
Map Service – Output image format choices

- **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

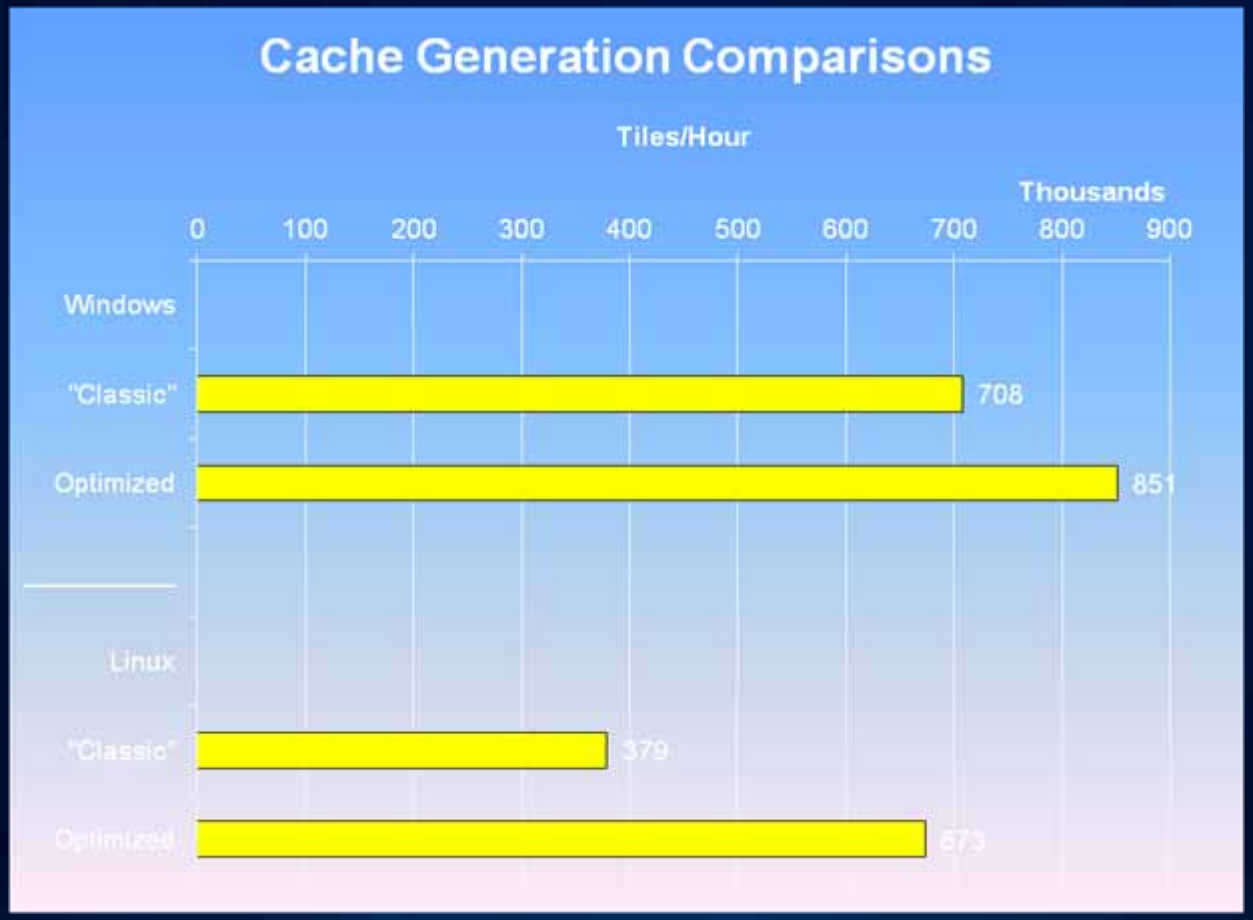
Map Service – Source map document optimizations, scale dependency

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



Performance Factors: ArcGIS Server Services

Map Service – Cache generation



- 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 (new at 10.0)
- 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: Machine Architecture

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***

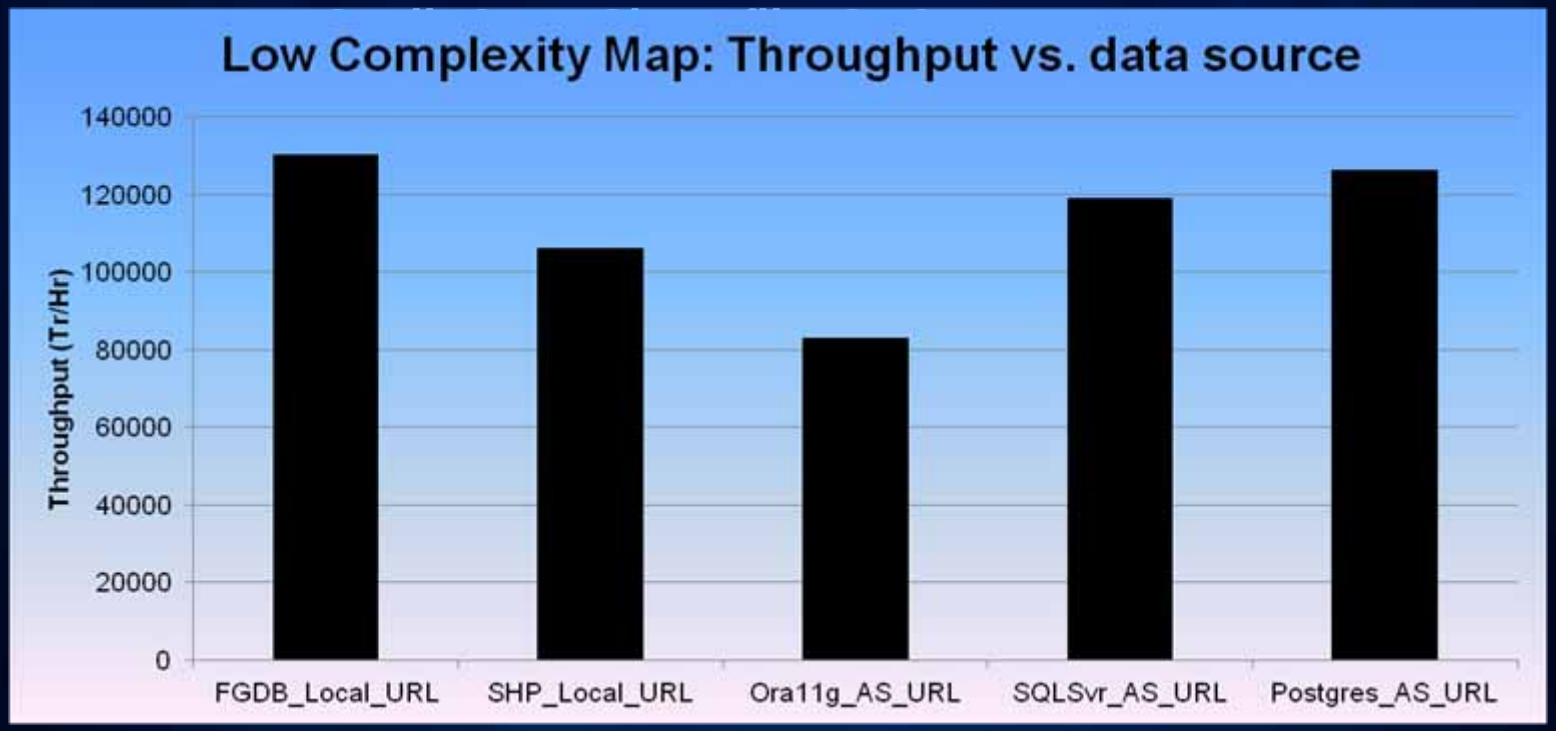
$$\#CPU_t = \frac{ST_b \times TH_t \times 100}{3600 \times \%CPU_t} \times \frac{SpecRatePerCPU_b}{SpecRatePerCPU_t}$$

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

Performance Factors: Data Sources

Select data storage format that provides optimal performance

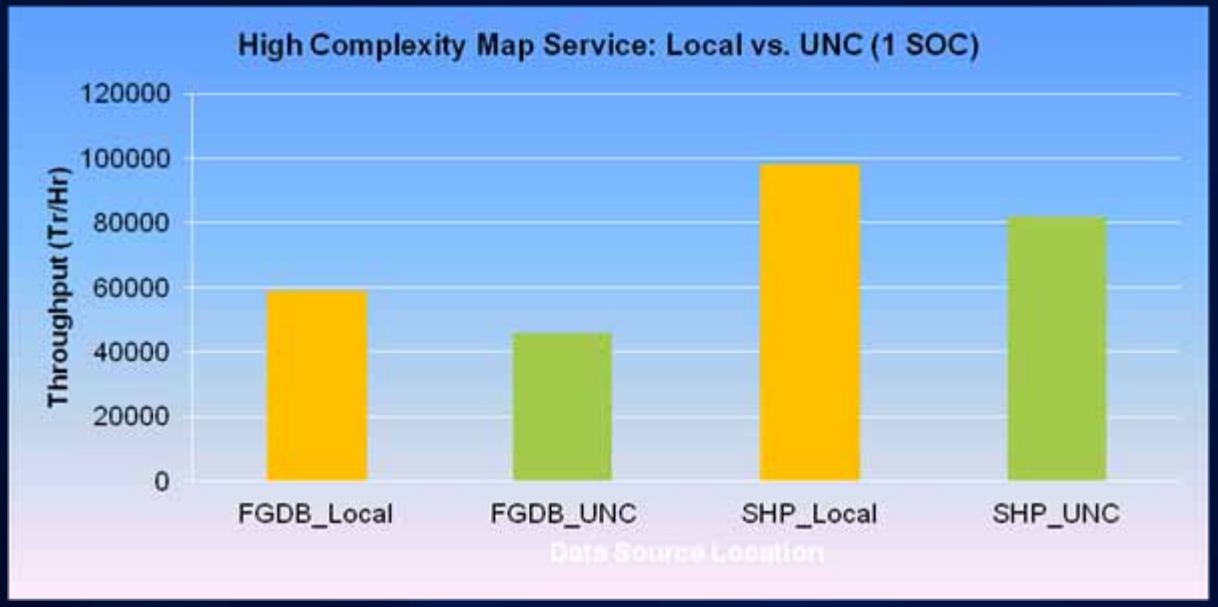
Data storage format



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**

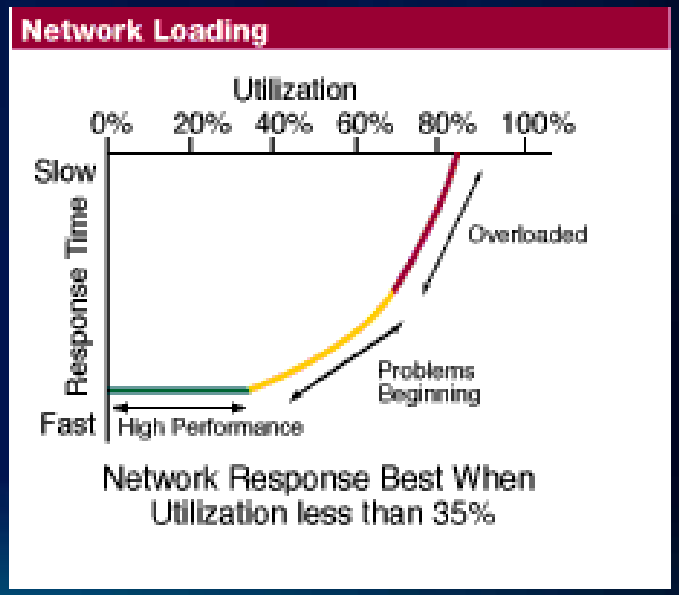
Non- optimal DBMS may be a source of significant performance degradation

Performance Factors: ArcGIS Server Framework

Web Service Components – Web Service Handlers

- **Under high load, easily overloaded by MIME data (REST/SOAP returning MIME)**
  - **Network saturation**
    - **72 cores can approach 40% Gigabit utilization**
    - **Network I/O (full-duplex)**
  - **CPU utilization**

- **Use multiple Web Instances**



Performance Factors: ArcGIS Server Framework

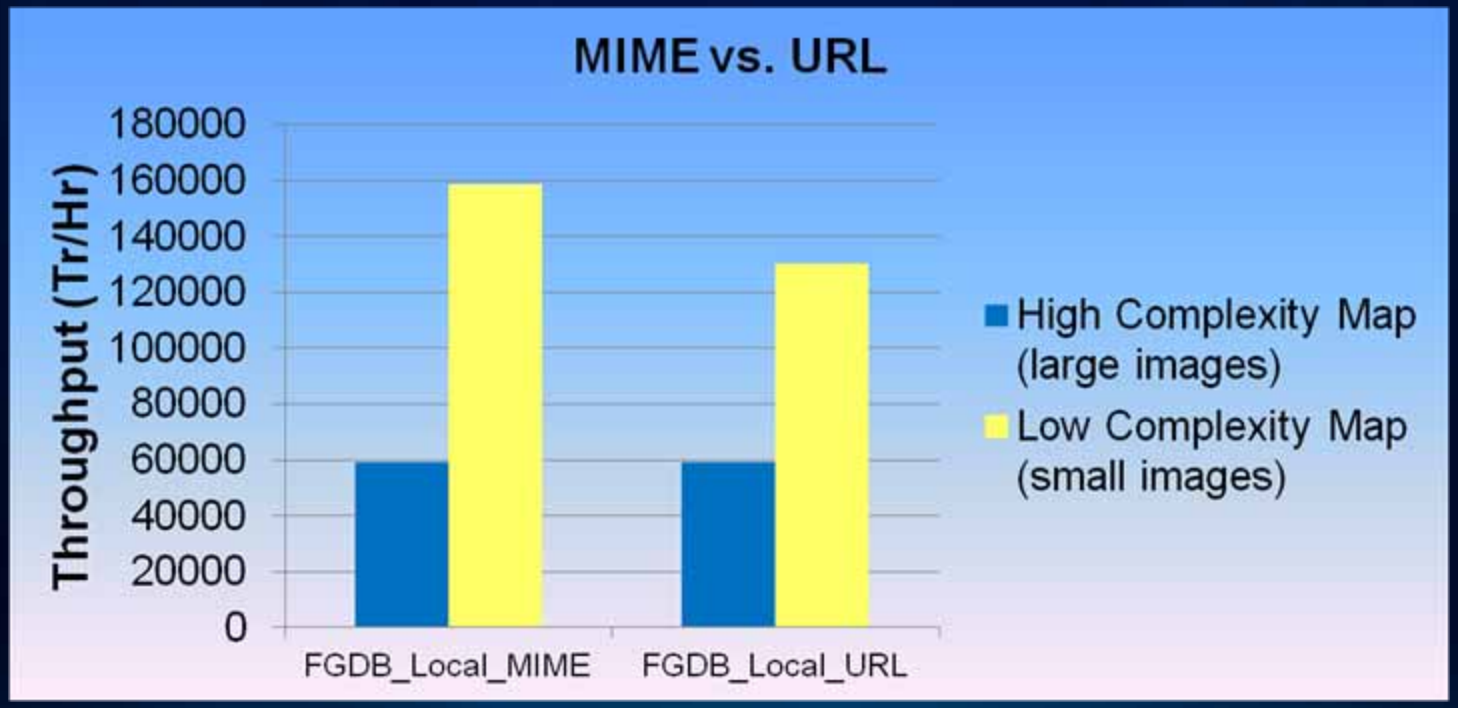
Web Service Components – Virtual Directories

- **Under high load, virtual output directory bottlenecks**
  - **Network saturation**
  - **Disk performance**
- **Use multiple output directories on different machines.**

Performance Factors: ArcGIS Server Framework

Web Service Components - MIME vs. URL return type

- For serving maps, MIME can scale better than URL
- Disk/UNC shares often bottlenecks before network bandwidth



- MIME performs better with smaller images

Performance Factors: ArcGIS Server Framework

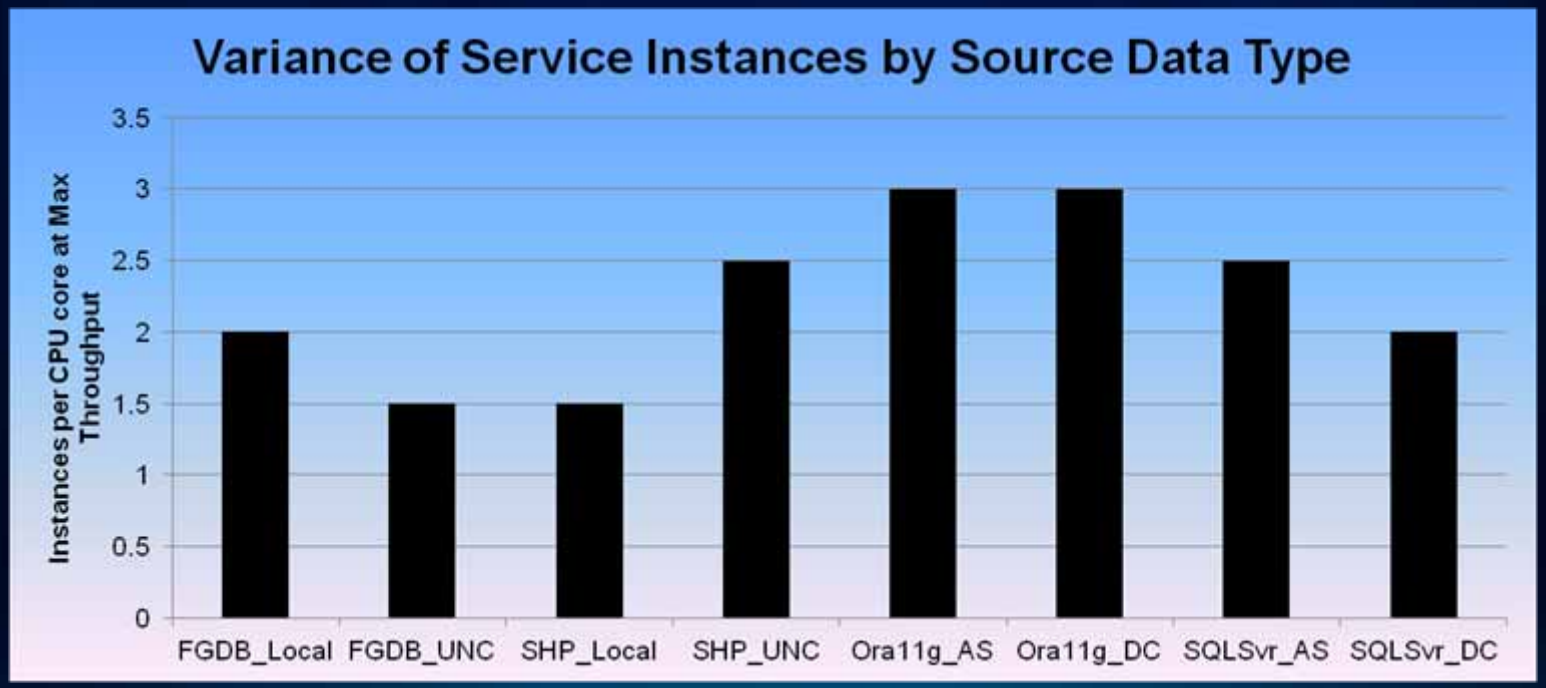
Web Services - Security

- **SSL**
- **LSASS**
  - http://support.esri.com/index.cfm?fa=knowledgebase.techarticles.articleShow&d=32620
- **Web Tier Security**
  - **User/Role Stores**
  - **Token Server**

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.



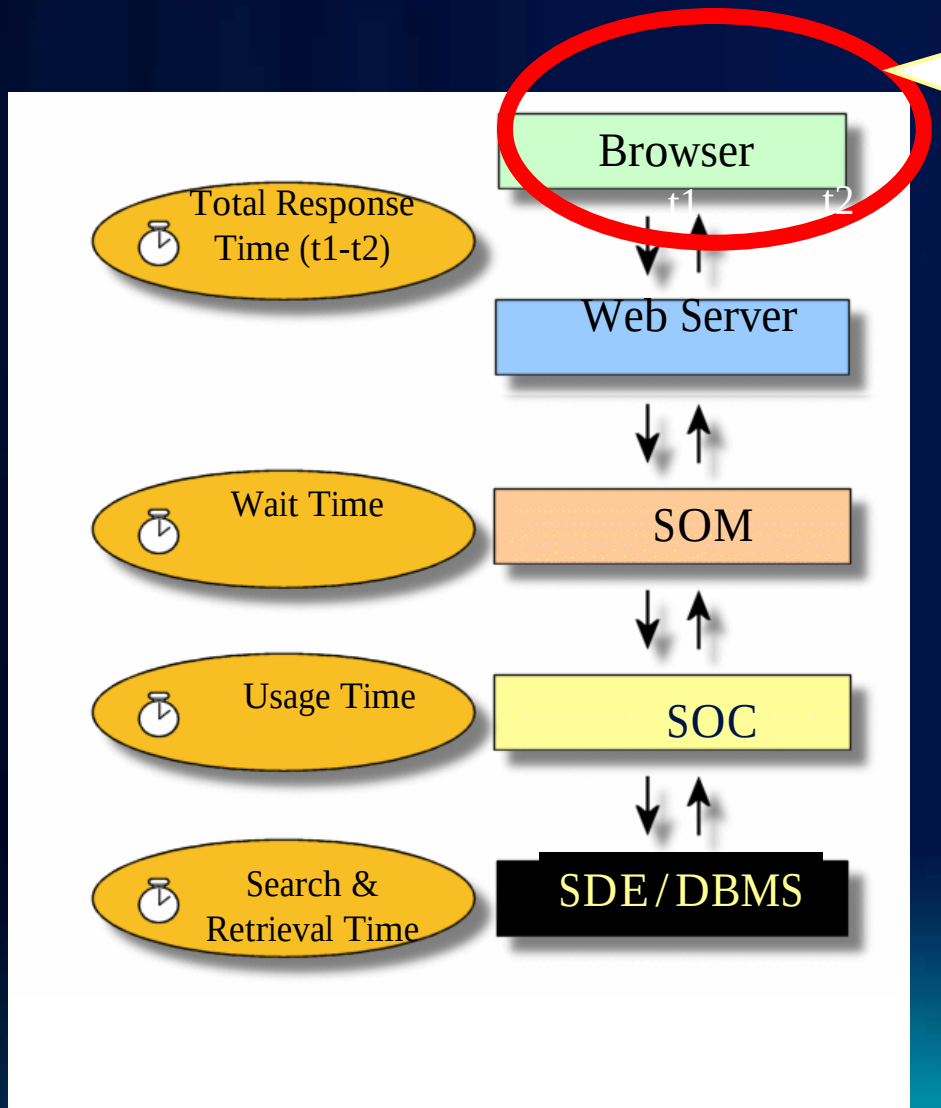
Tuning Primer

Performance tuning

- Benefits
  - Improved performance - user experience
  - optimize the use of critical resources – scalability
- Tools
  - Fiddler
  - Mxdperfstat
  - Map Service Publishing Toolbar
  - DBMS trace

Tuning Primer

Profile user transaction response time

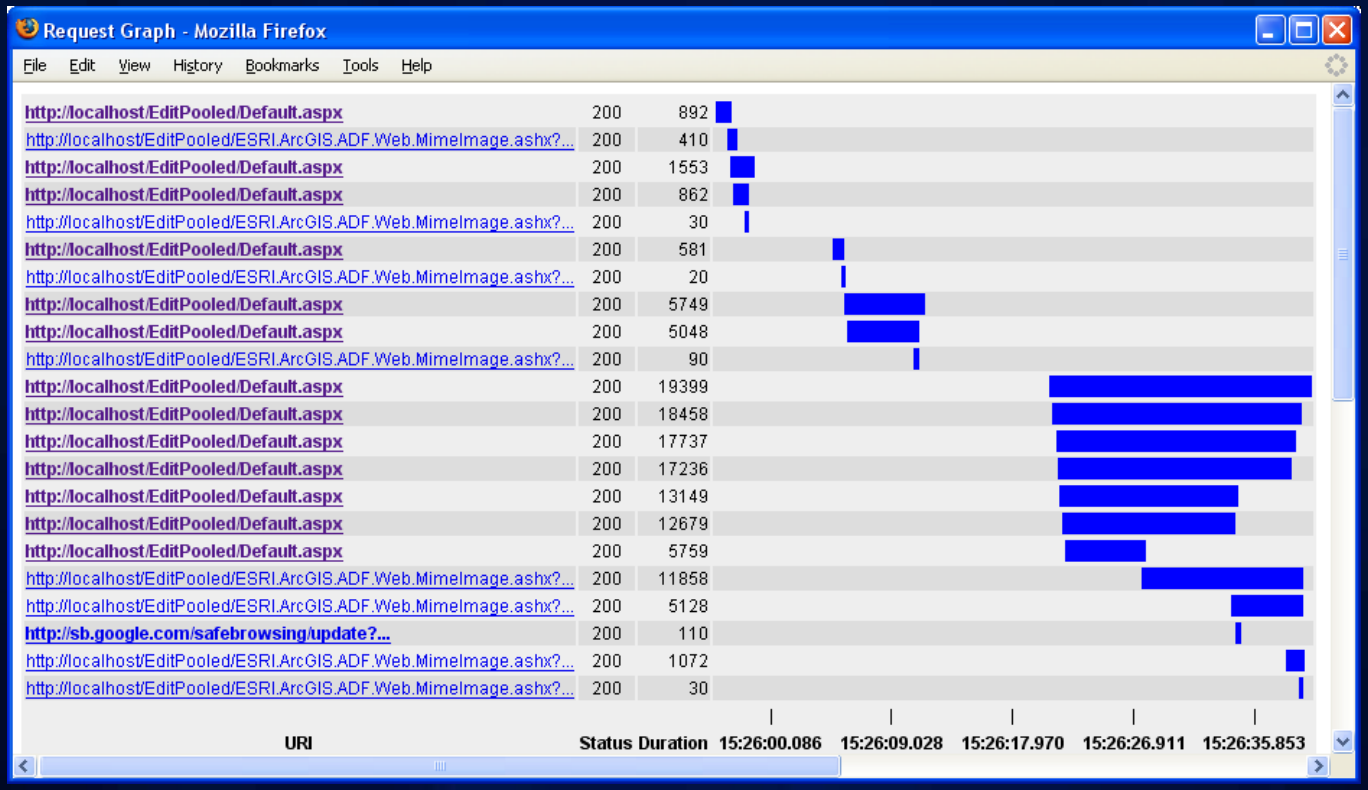


A test is executed at the web browser.

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

Tuning Primer

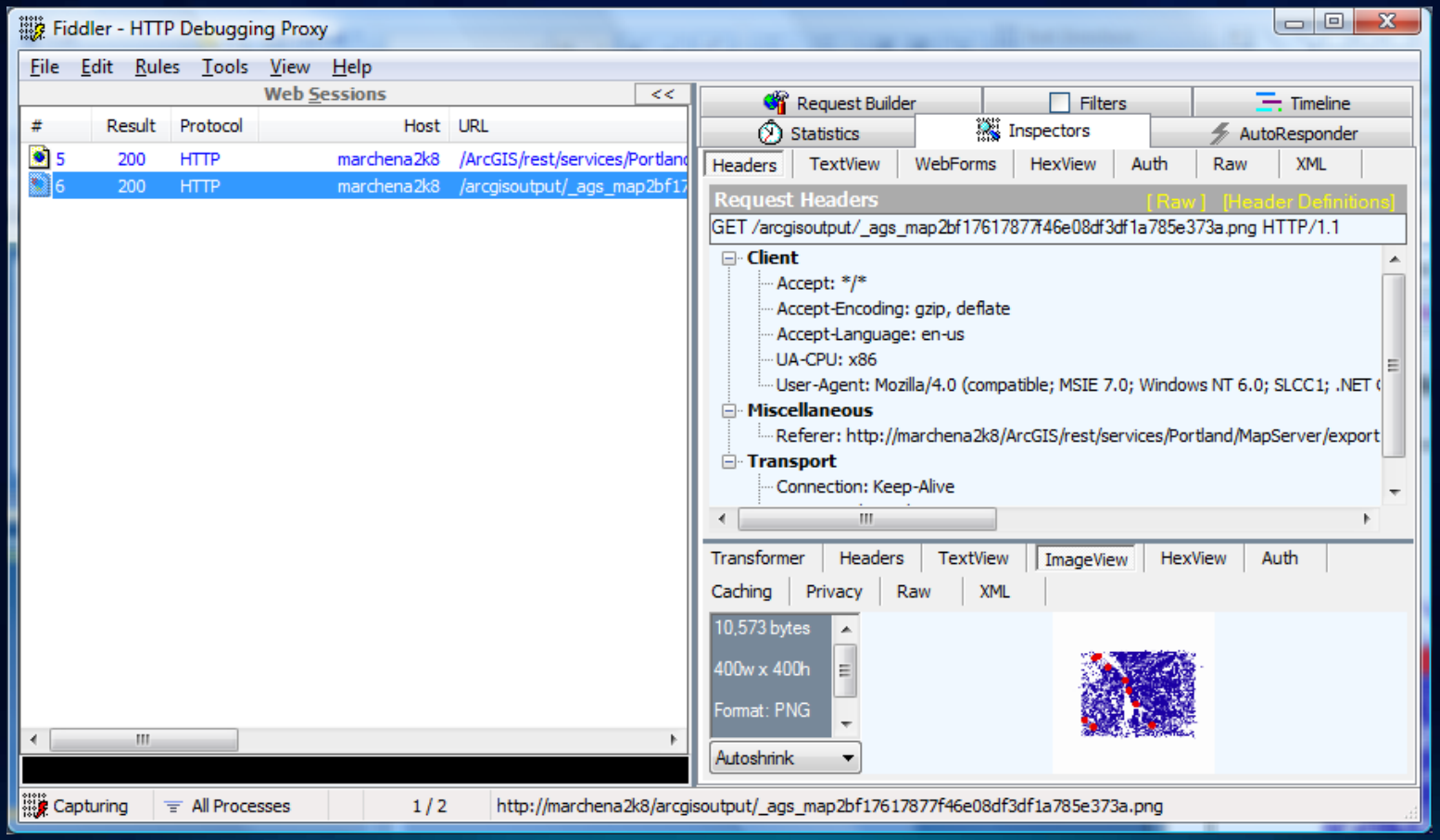
Web diagnostic tools: Fiddler, Tamperdata, Yslow



Tuning Primer

Web diagnostic tools: Fiddler

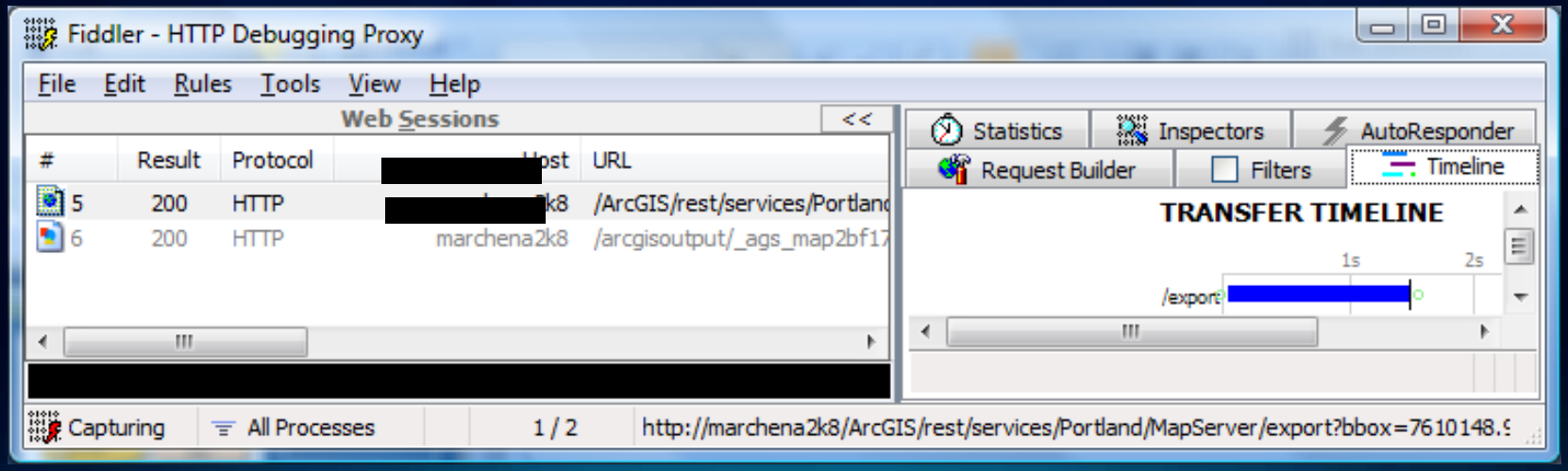
- Can validate image returned



Tuning Primer

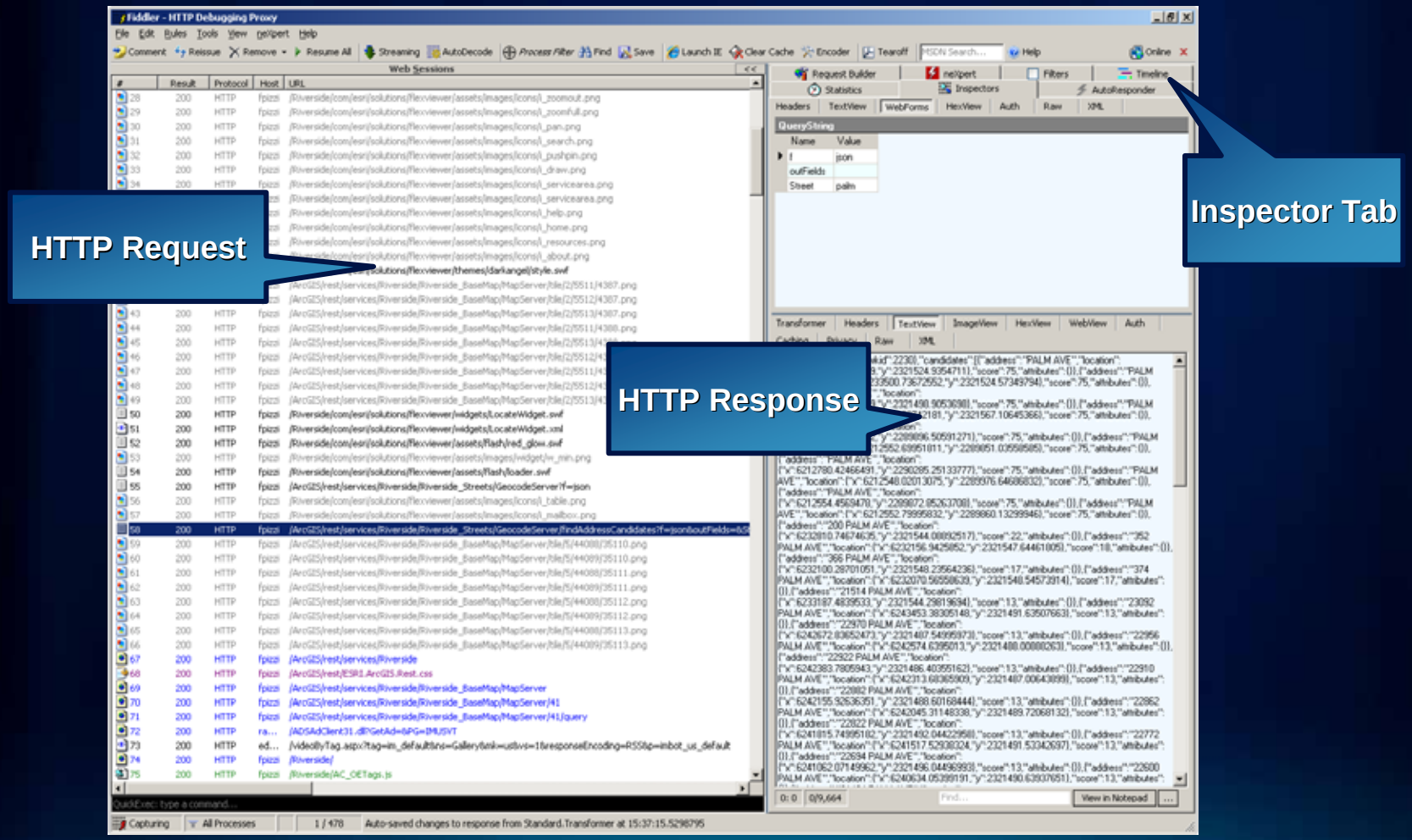
Web diagnostic tools: Fiddler

- 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



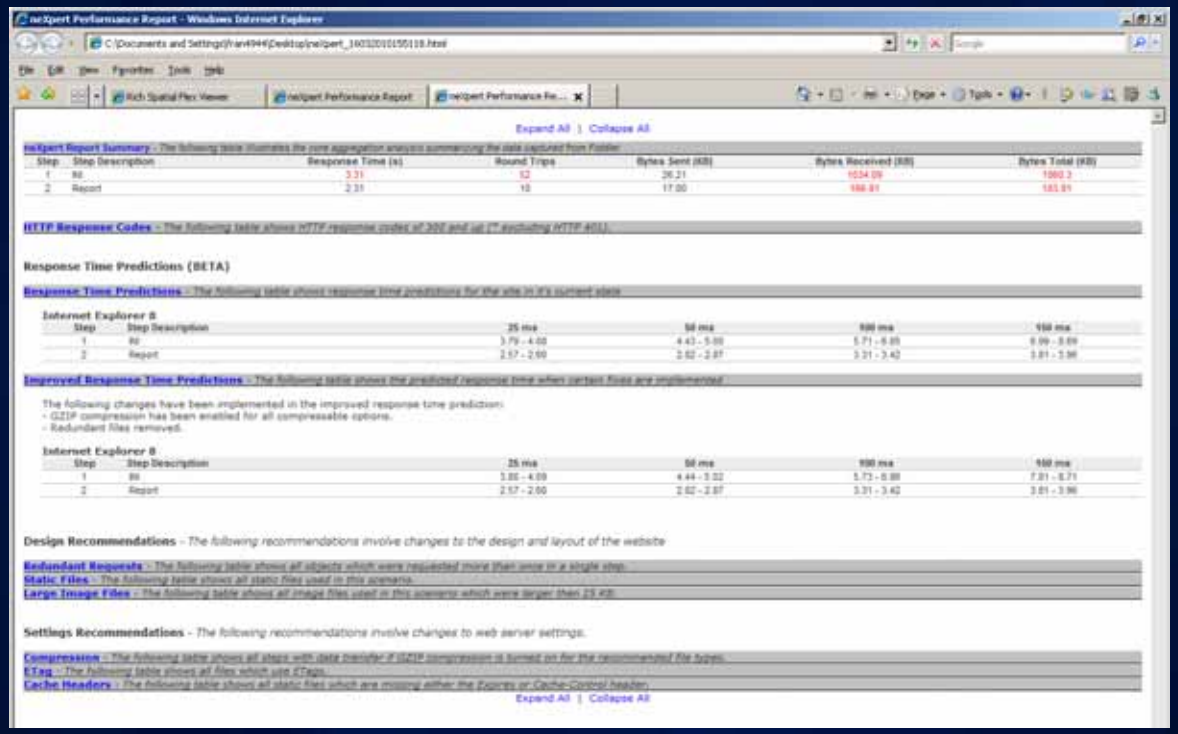
Tuning Primer

Web diagnostic tools: Fiddler



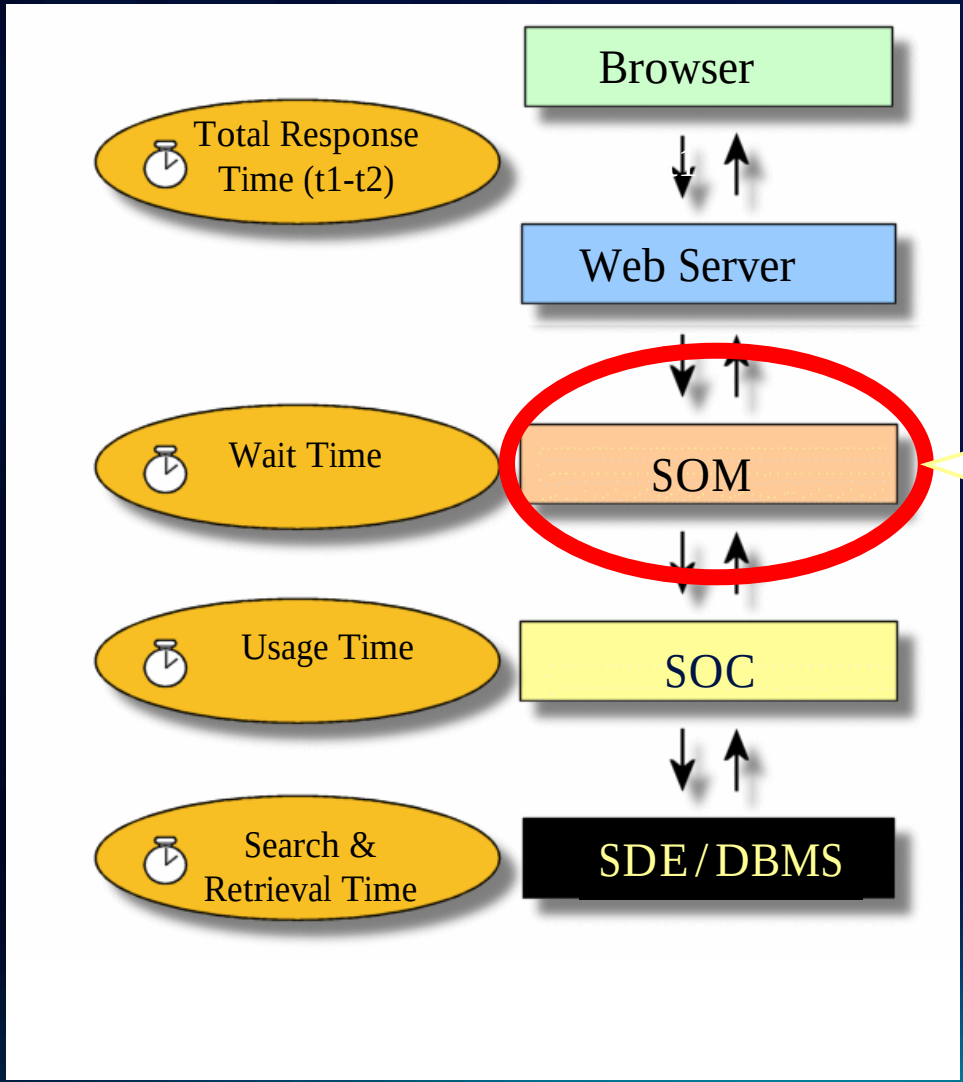
Tuning Primer

Web diagnostic tools: Fiddler (NeXpert Report)



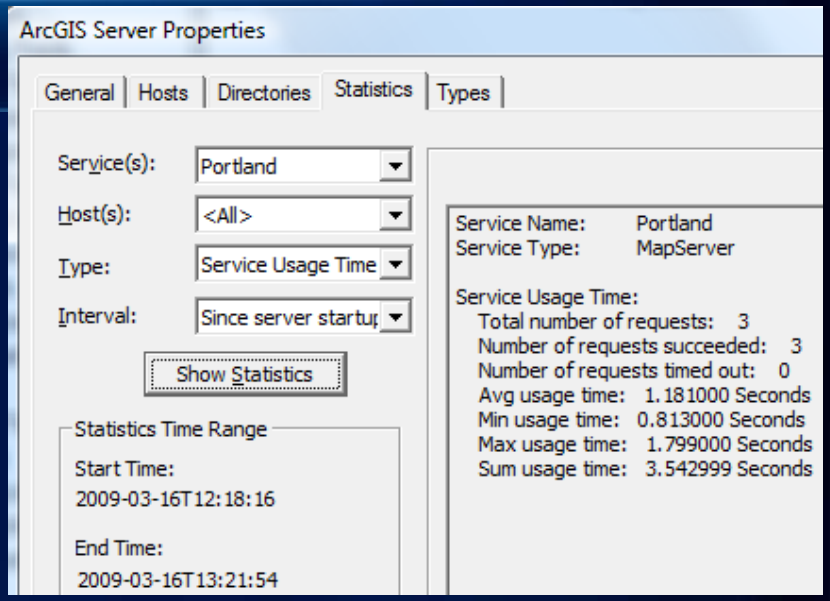
Tuning Primer

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



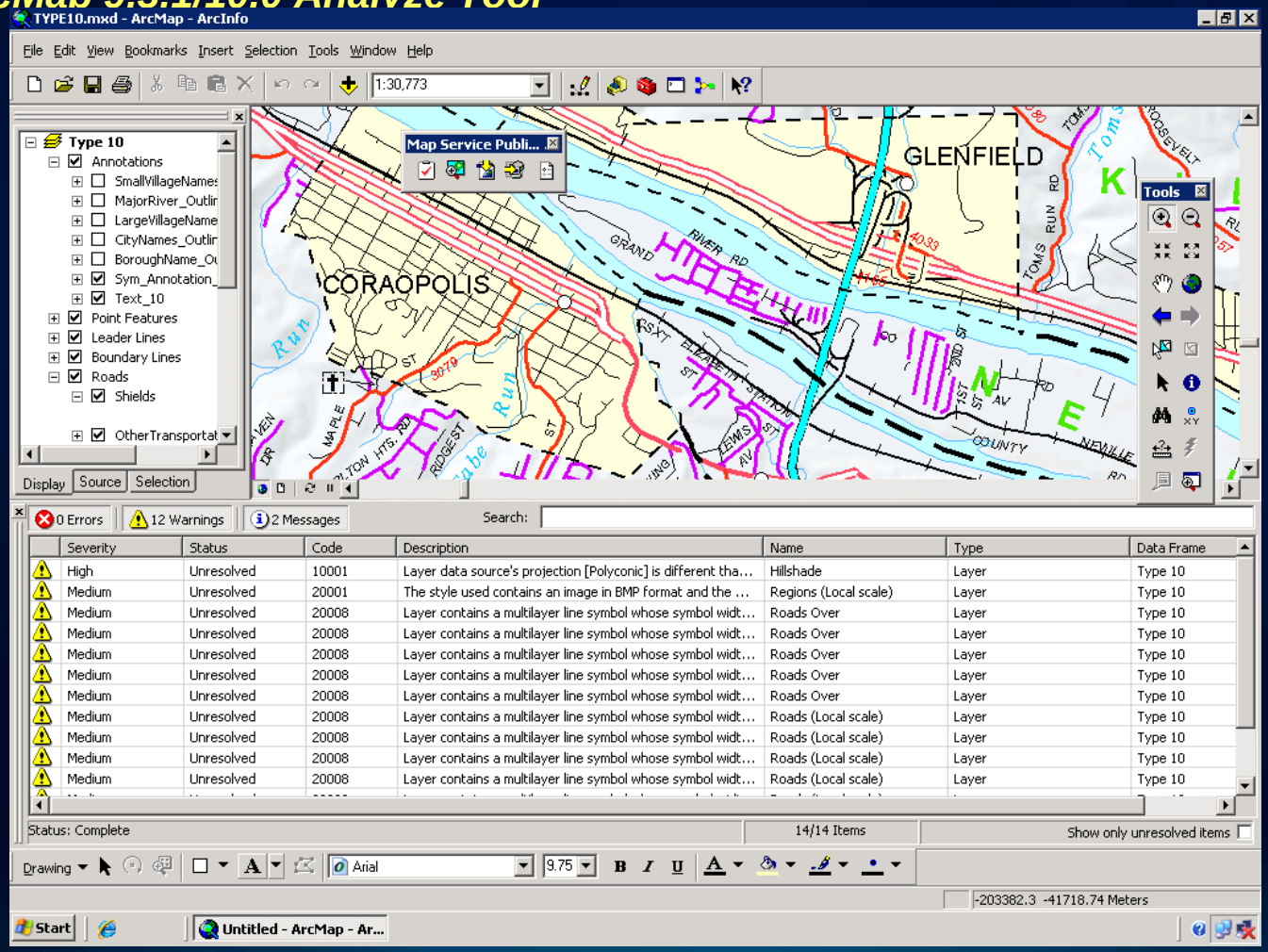
- **ArcCatalog**

- **Detailed log - set to verbose**

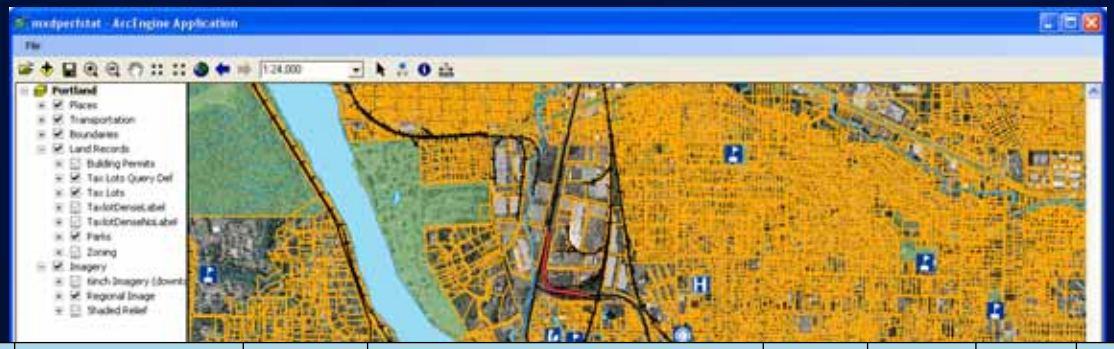

```
<Msg time="2009-03-16T12:23:22" type="INFO3" code="103021" target="Portland.MapServer" methodName="FeatureLayer.Draw" machine="myWebServer" process="2836" thread="3916" elapsed="0.05221">Executing query.</Msg>

<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



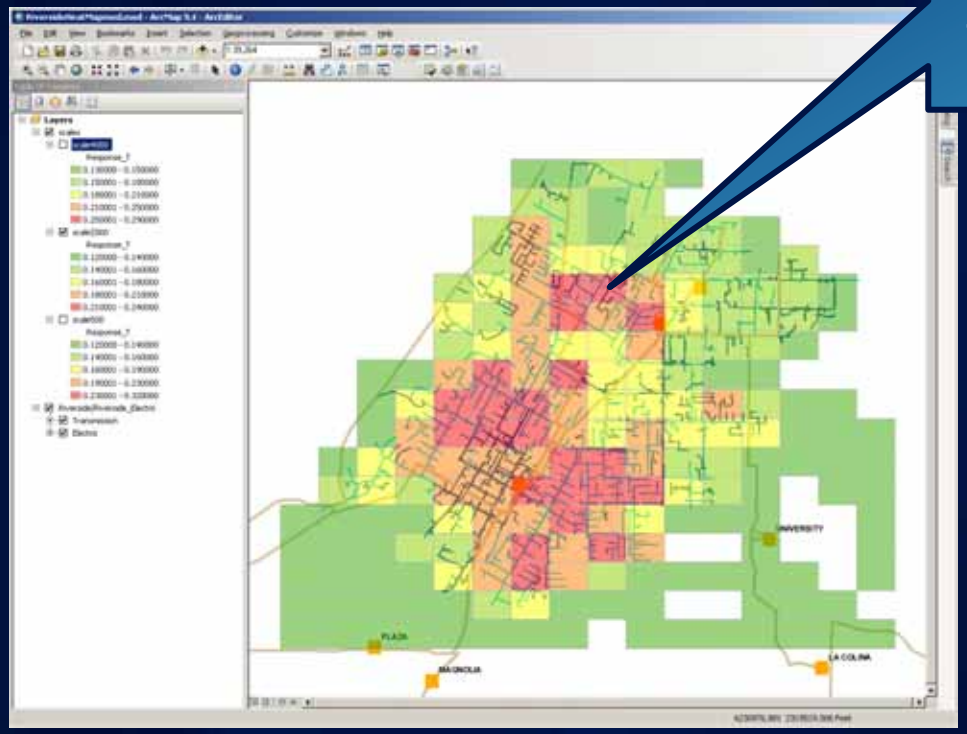
Tuning Primer
mxdperfstat



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

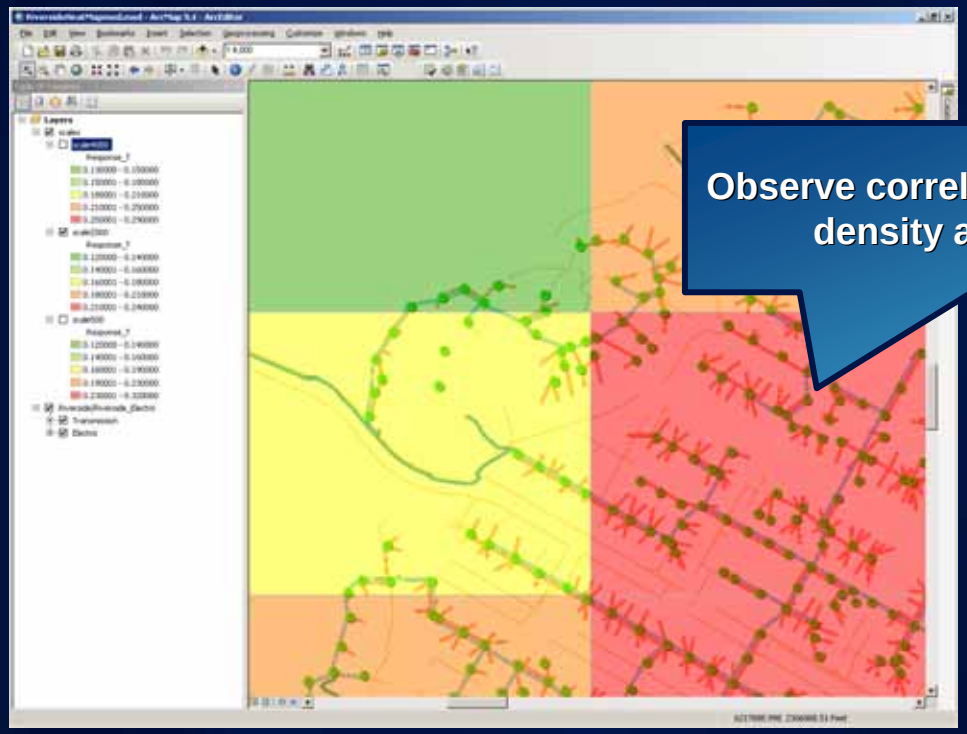
ArcGIS Services



Heat Map based on response times from ArcGIS Server

Tuning Primer

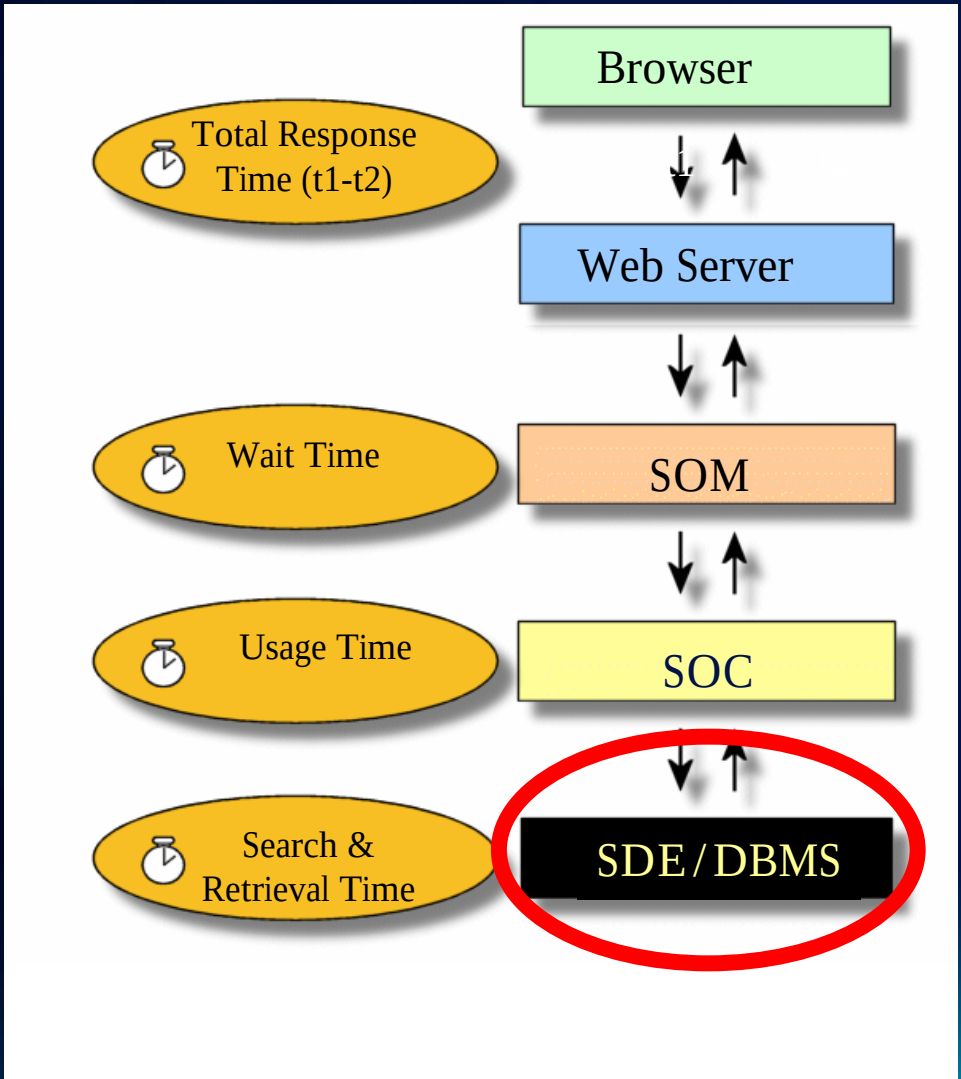
ArcGIS Services



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)

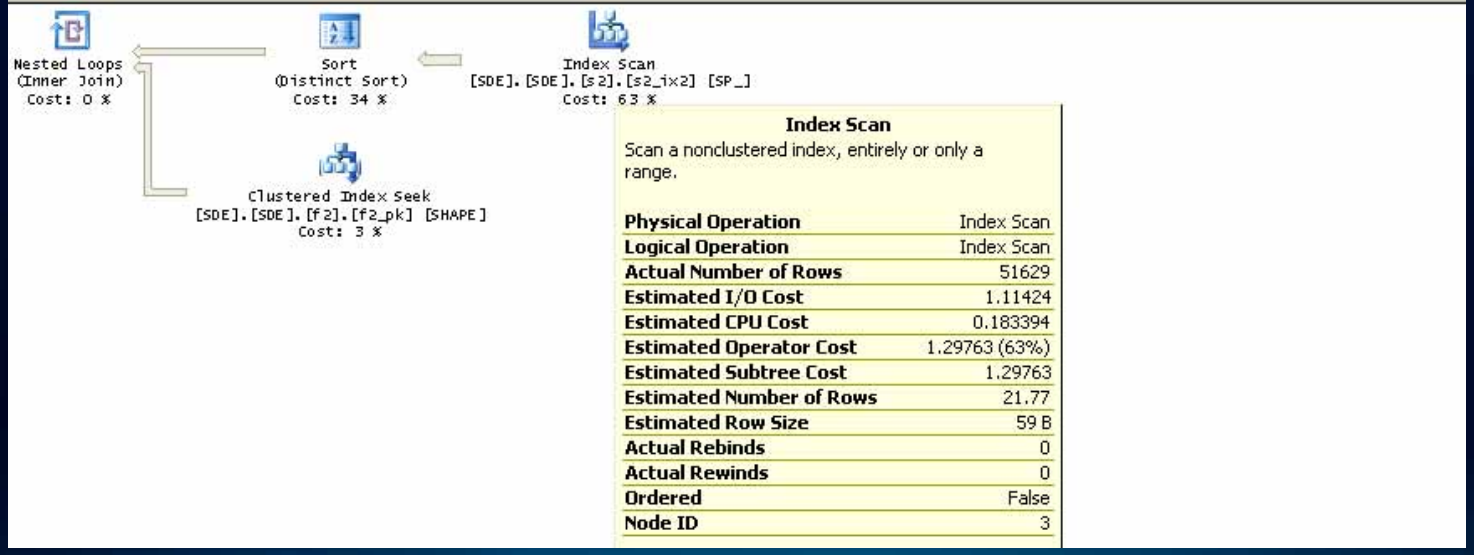
- Example (cost of physical reads):
  - Elapsed time = 13.02 sec
  - CPU = 0.85 sec
  - Disk = 2291 blocks
  - Wait event (db file sequential read)=11.69 sec
  - Rows fetched = 24175

Tuning Primer

Data Sources – SQL Profiler

ArcSDE_trace (ANDREWS2)

EventClass	Login...	Application...	TextData	CPU	Duration	RowCounts	Reads
Trace Start							
Showplan XML Statistics P...	sde	SDE:5932	<ShowPlanXML xmlns="http://schemas....				
SP:StmtCompleted	sde	SDE:5932	SELECT state_id,owner,creation_time...	10	0	1	2
Showplan XML Statistics P...	sde	SDE:5932	<ShowPlanXML xmlns="http://schemas....				
SP:StmtCompleted	sde	SDE:5932	SELECT lineage_name, time_last_modi...	0	0	1	2
Showplan XML Statistics P...	sde	SDE:5932	<ShowPlanXML xmlns="http://schemas....				
SP:StmtCompleted	sde	SDE:5932	SELECT S_.eminx,S_.eminy,S_.emaxx,S...	0	0	0	0
Showplan XML Statistics P...	sde	SDE:5932	<ShowPlanXML xmlns="http://schemas....				
SP:StmtCompleted	sde	SDE:5932	SELECT S_.eminx,S_.eminy,S_.emaxx,S...	521	2624	36251	11...



Testing

Testing

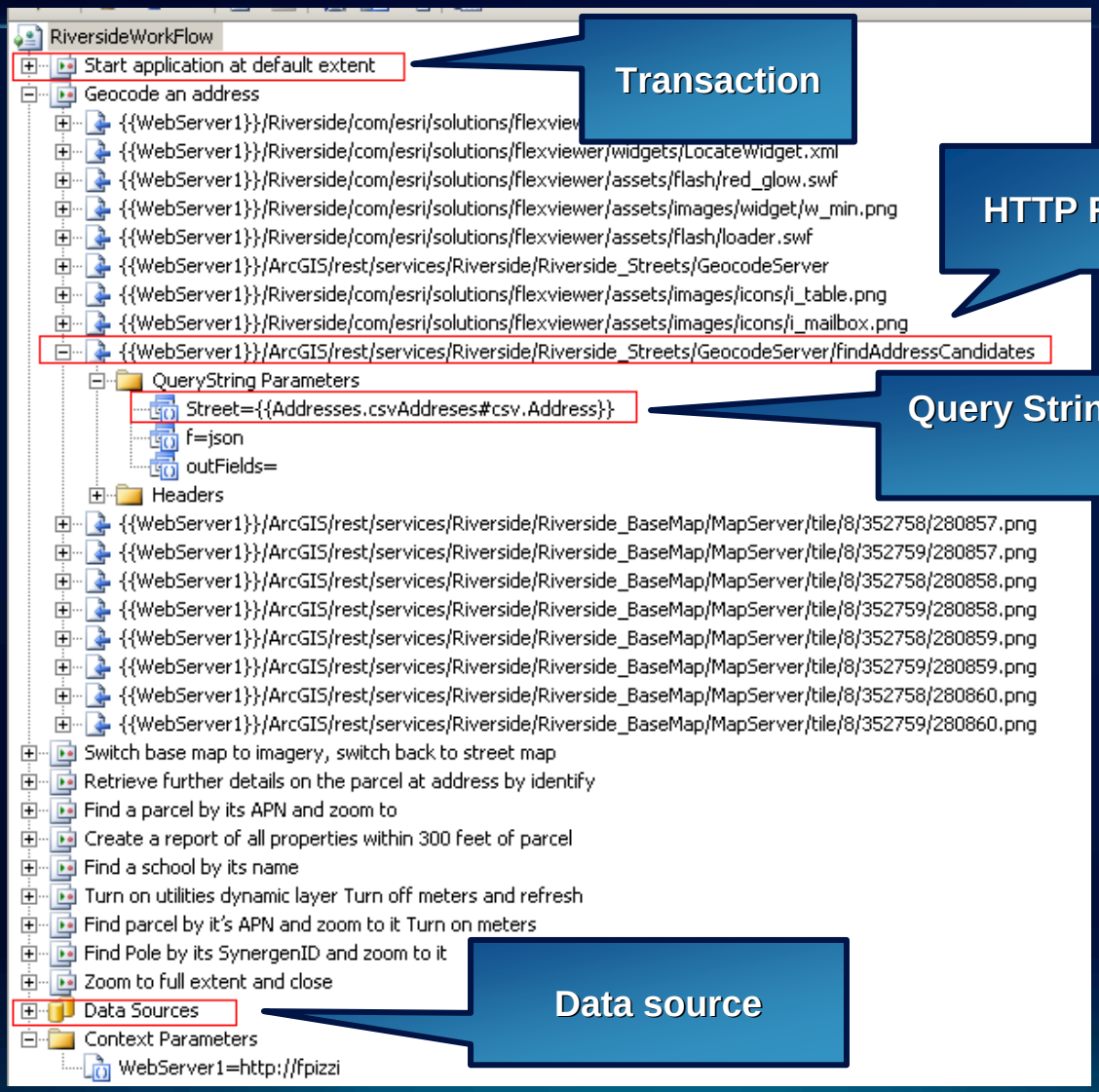
Key Tasks

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

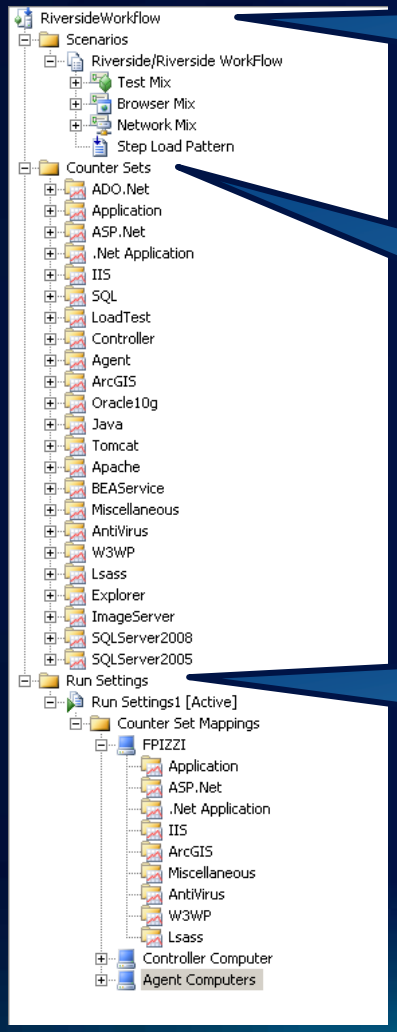
Visual Studio Quick Introduction

Visual Studio Quick Introduction

Visual Studio Quick Introduction - WebTest



Visual Studio Quick Introduction – Load Test

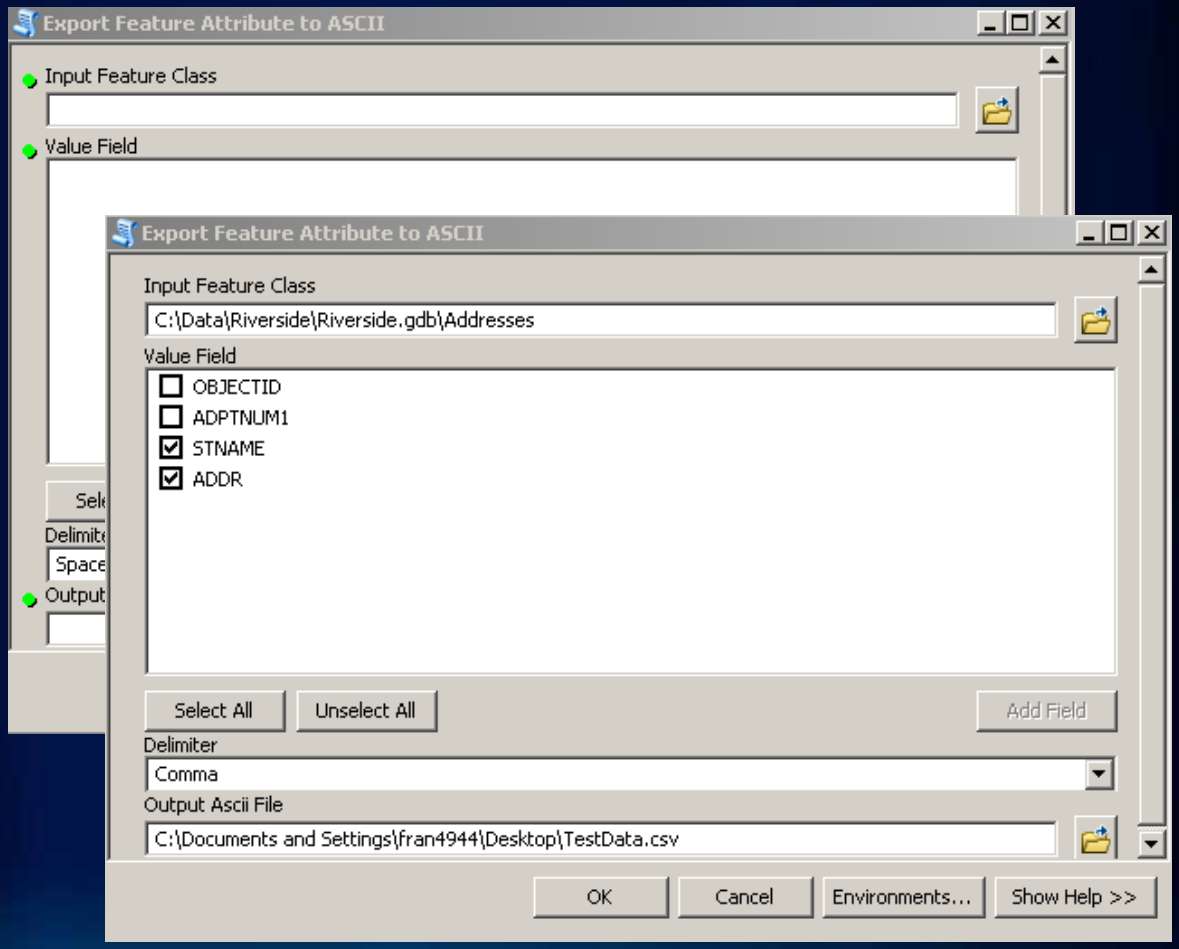


Scenarios:
Test Mix (WebTest or Unit Test),
Browser Mix,
Network Mix,
Step Loads

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

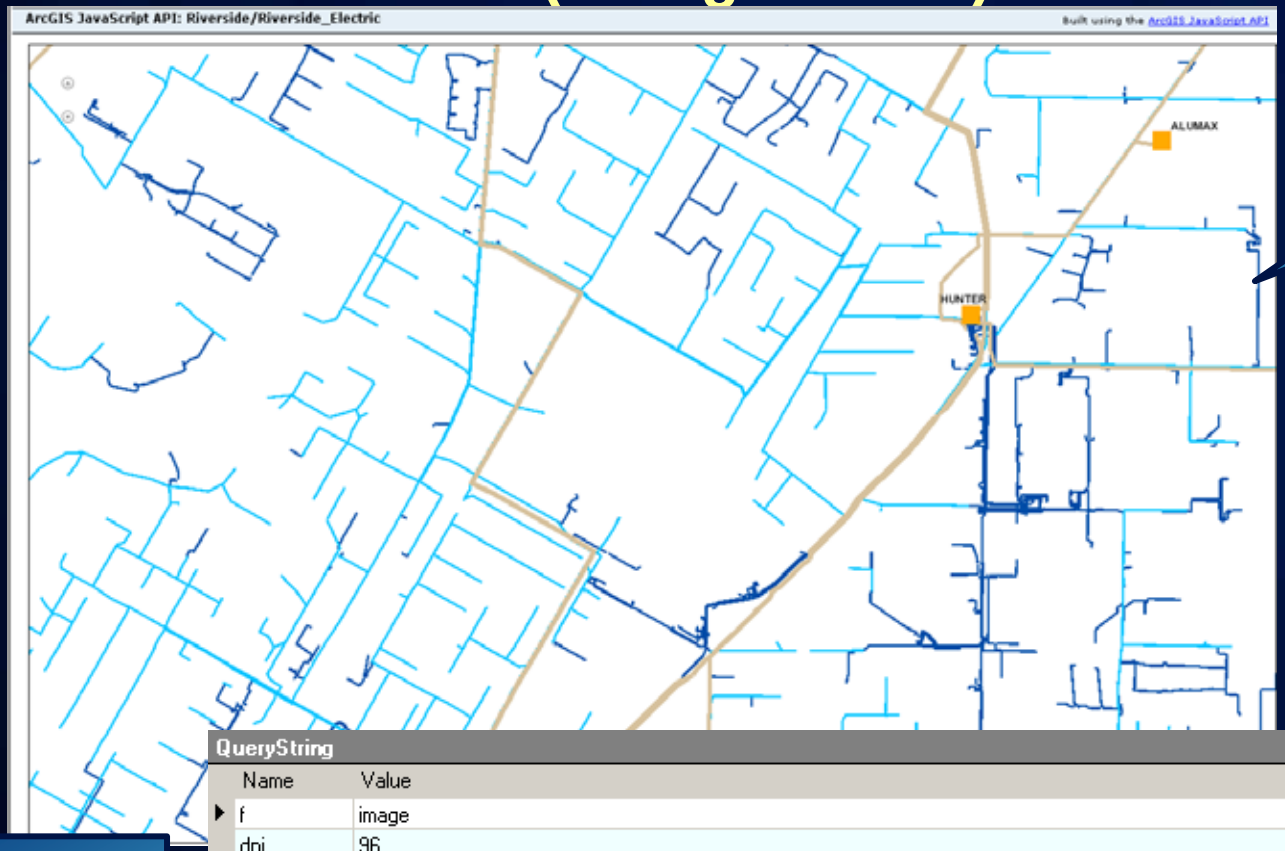
Run Settings:
Counter Set Mappings – Machine metrics
Test duration

Testing
Test Data – Attribute Data



Testing

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

Testing
Test Data – Bbox

One simple example of Python script to generate Bboxes

```python
"""
Generate Bboxes from selected Riverside extent from ...
Note* Image size=1222,782
"""
__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)
```

Testing Tools

Testing - Selecting Web 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

Validate Test

Tips and Tricks

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

Results Validation

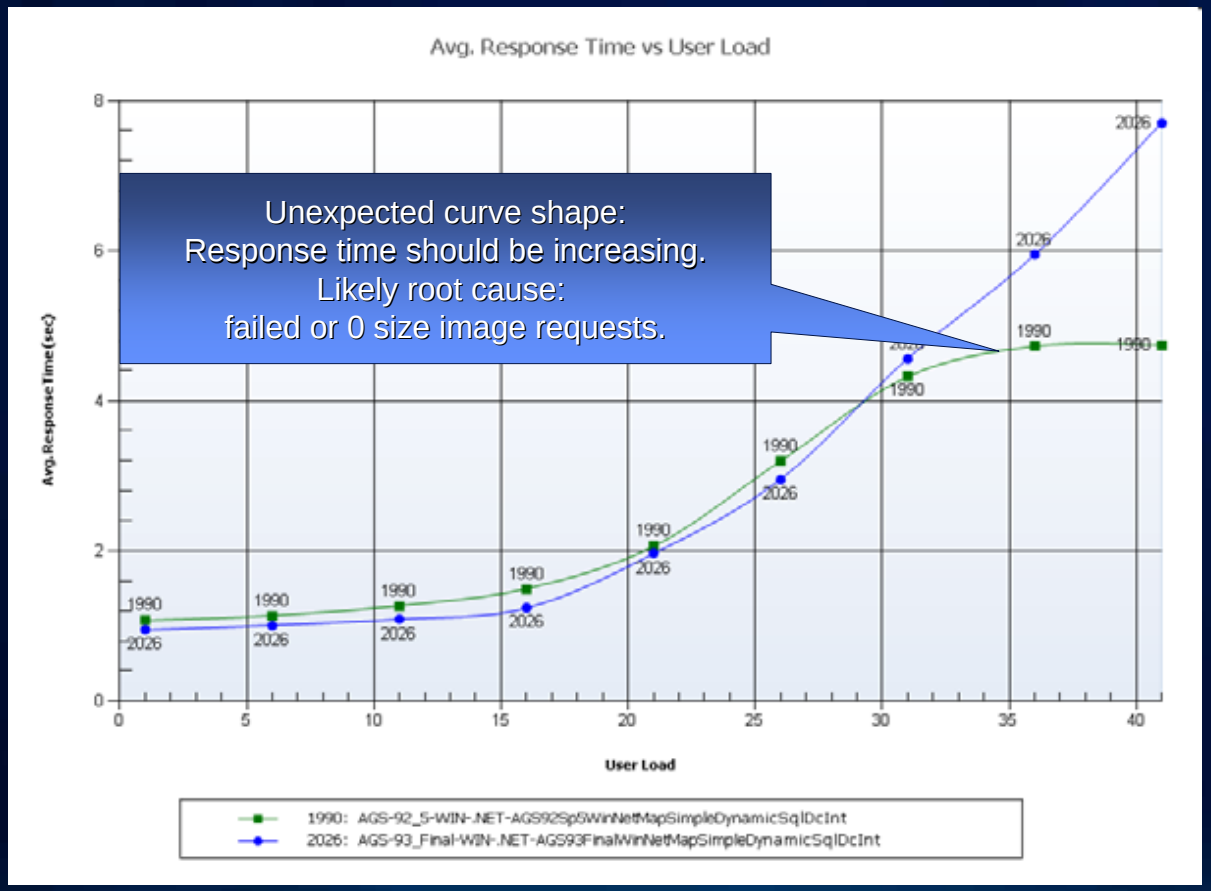
- Lack of errors does not validate a test
  - Requests may succeed but return zero size image
  - Spot check request response content size

Results Validation - Reporting

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

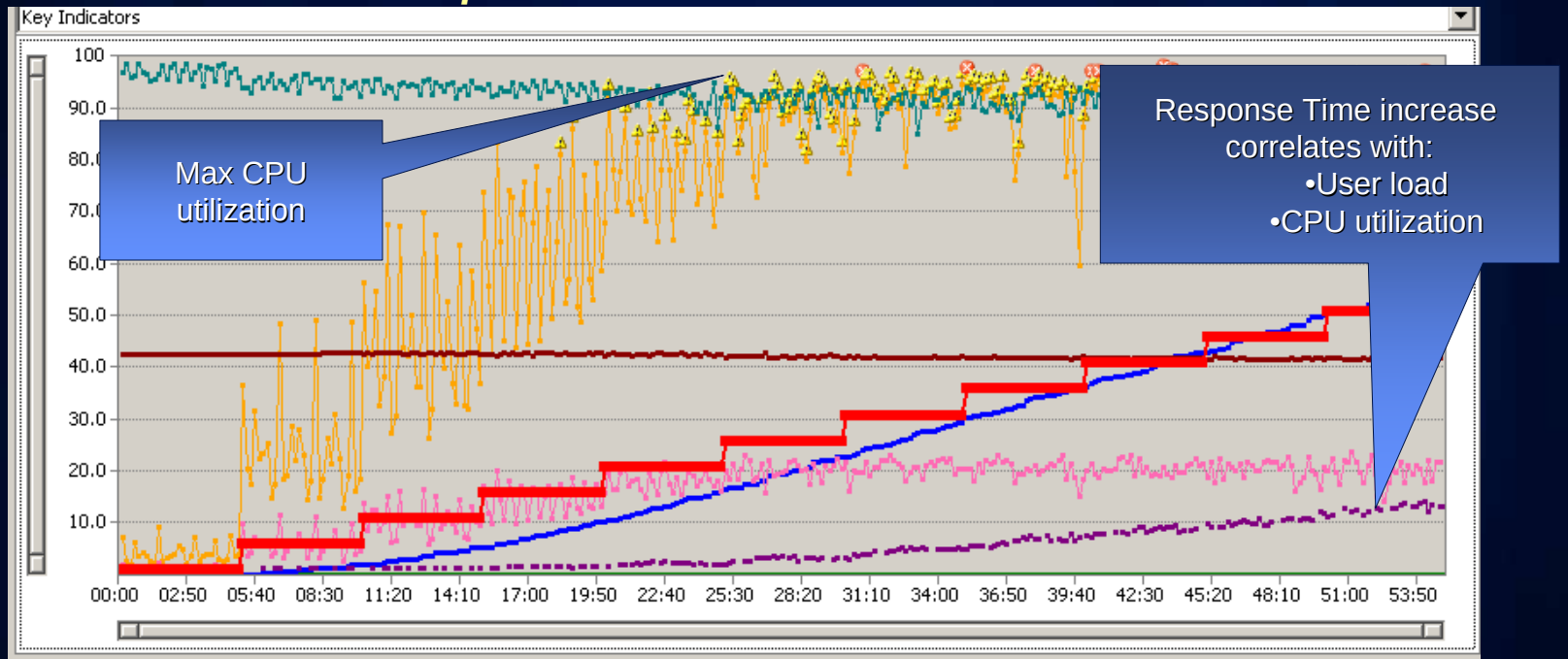
Results Validation

Incorrect CPU and Response Time Correlation



Results Validation

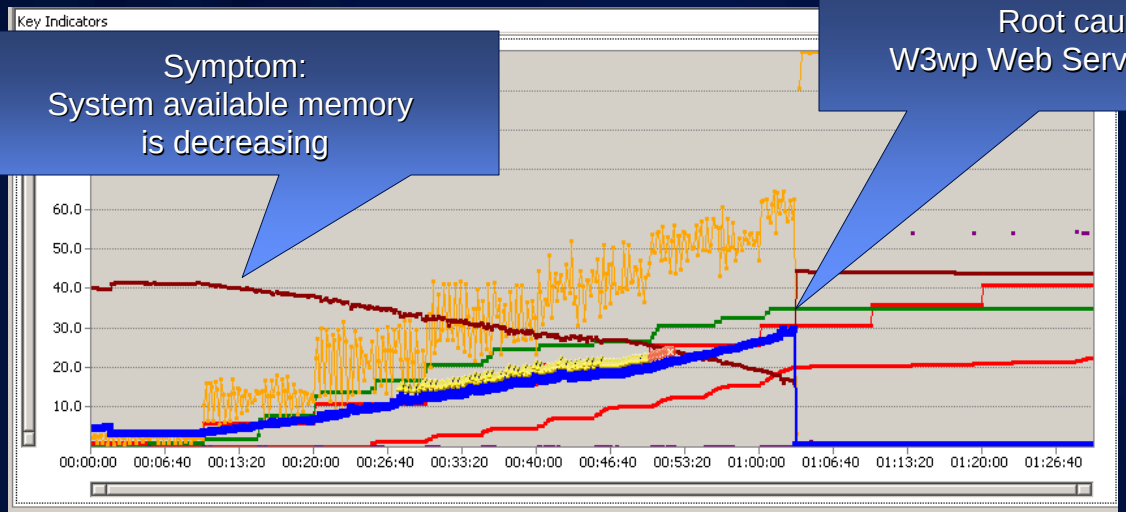
Correct CPU and Response Time Correlation



Counter	Instance	Category	Computer	Color	Range	Min	Max	Avg
☑ User Load	_Total	LoadTest:Scenar	ESLWS9		100	1	51	26
☑ Failed Tests	_Total	LoadTest:Test	ESLWS9		0	0	0	0
☑ Passed Tests	_Total	LoadTest:Test	ESLWS9		1,000	0	568	223
☑ Avg. Response Time	_Total	LoadTest:Transa	ESLWS9		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

Results Validation

Test failure due to w3wp memory bottleneck

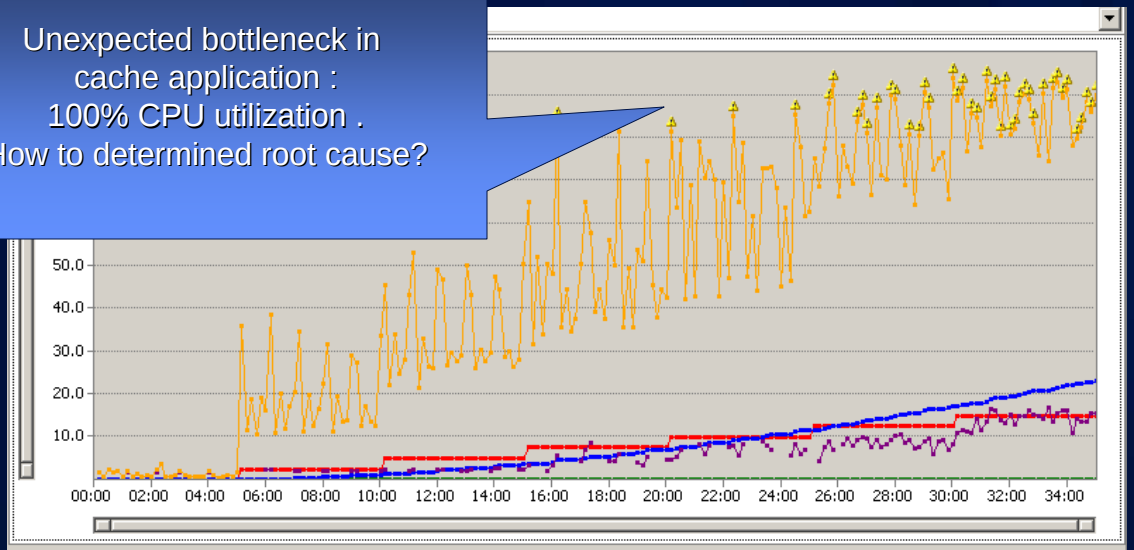


Counter	Instance	Category	Computer	Color	Range	Min	Max	Avg
☑ User Load	_Total	LoadTest:Scenar	ESLIBMBL4		100	1	41	21
☑ Failed Tests	_Total	LoadTest:Test	ESLIBMBL4		1,000	0	226	99
☑ Passed Tests	_Total	LoadTest:Test	ESLIBMBL4		100	0	35	23
☑ Avg. Response Time	_Total	LoadTest:Transa	ESLIBMBL4		1,000	0.31	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

Results Validation

Unexpected CPU utilization while accessing cache services

Unexpected bottleneck in
cache application :
100% CPU utilization .
How to determined root cause?



Counter	Instance	Category	Computer	Color	Range	Min	Max	Avg
User Load	_Total	LoadTest:Scena	ESLWS9		1,000	1	151	76
Failed Tests	_Total	LoadTest:Test	ESLWS9		0	0	0	0
Passed Tests	_Total	LoadTest:Test	ESLWS9		10,000	0	2,338	753
Avg. Response Time	_Total	LoadTest:Trans	ESLWS9		10	0.19	1.71	0.87
% Processor Time	_Total	Processor	ESLSRV9		100	0.59	94.1	47.8

Reporting Results

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**

Configuration validation

- Application stack is not properly configured and tuned
- Single error in a test can lead to a cascade of failing requests
- Test client machine is bottleneck
  - CPU
  - Memory
  - Networks (cache test in particular)

Contact us

- **Professional Services**
  - http://www.esri.com/services/professional-services/implementation/request-services.html
  - profservices@esri.com
- **Andrew Sakowicz, asakowicz@esri.com**
- **Your account manager**

Questions?

Esri Federal User Conference

GIS: The Geographic Approach for the Nation

Demo Related and Supplemental Slides

Jmeter

Testing with JMeter

- Riverside Electric MSD Service
  - Testing REST Export Map
- http://jakarta.apache.org/jmeter/

Testing with JMeter – ArcGIS Server REST

Build HTTP Request

RiversideElectric.jmx (C:\Documents and Settings\fran4944\Desktop\DevSummit2010\Jmeter\RiversideEle

File Edit Run Options Help

0 / 1

- RiversideElectric
  - HTTP Request Defaults
  - CSV Data
  - Aggregate Graph
  - Validator
    - Validator
    - REST Results
    - Aggregate Report
    - View Results Tree
  - 1 User
    - 1 User
  - 2 Users
    - 2 Users
  - 3 Users
    - 3 Users
  - 4 Users
    - 4 Users
- WorkBench

HTTP Request Defaults

Name: HTTP Request Defaults

Comments:

Web Server

Server Name or IP: fpizzi **Port Number:**

Timeouts (milliseconds)

Connect: **Response:**

HTTP Request

Protocol (default http): **Content encoding:**

Path: /ArcGIS/rest/services/Riverside/Riverside_Electric/MapServer/export?

Send Parameters With the Request:

Name:	Value	Encode?	Include Equ...
f	image	☐	☑
dpi	96	☐	☑
transparent	true	☐	☑
format	png8	☐	☑
size	1222,782	☐	☑

Add Delete

☐ Retrieve All Embedded Resources from HTML Files

Testing with JMeter– Test Data from CSV

Set Test Data

RiversideElectric.jmx (C:\Documents and Settings\fran4944\Desktop\DevSummit2010\Jmeter\RiversideElectric.j

File Edit Run Options Help

0 / 1

- RiversideElectric
  - HTTP Request Defaults
  - CSV Data
  - Aggregate Graph
  - Validator
    - Validator
    - REST Results
    - Aggregate Report
    - View Results Tree
  - 1 User
    - 1 User
  - 2 Users
    - 2 Users
  - 3 Users
    - 3 Users
  - 4 Users
    - 4 Users
- WorkBench

CSV Data Set Config

Name: CSV Data

Comments:

Configure the CSV Data Source

Filename:	RiversideElectric.csv
File encoding:	
Variable Names (comma-delimited):	MinX,MinY,MaxX,MaxY
Delimiter (use '\t' for tab):	,
Allow quoted data?:	False
Recycle on EOF ?:	True
Stop thread on EOF ?:	False
Sharing mode:	All threads

Testing with JMeter – Set Thread Properties

RiversideElectric.jmx (C:\Documents and Settings\fran4944\Desktop\DevSummit2010\Jmeter\RiversideElectric.jmx) - Apache JMeter (2.3.4 r785646)

File Edit Run Options Help

0 / 1

- RiversideElectric
  - HTTP Request Defaults
  - CSV Data
  - Aggregate Graph
  - Validator
    - Validator
    - REST Results
    - Aggregate Report
    - View Results Tree
  - 1 User
    - 1 User
  - 2 Users
    - 2 Users
  - 3 Users
    - 3 Users
  - 4 Users
    - 4 Users
  - WorkBench

HTTP Request

Name: 1 User

Comments:

Web Server

Server Name or IP: Port Number:

Timeouts (milliseconds)

Connect: Response:

HTTP Request

Protocol (default http): Method: GET Content encoding:

Path:

☑ Redirect Automatically ☐ Follow Redirects ☑ Use KeepAlive ☐ Use multipart/form-data for HTTP POST

Send Parameters With the Request:

Name:	Value	Encode?	Include Equ...
bbox	${MinX},${MinY},${MaxX},${MaxY}	☑	☑

Add Delete

Send Files With the Request:

File Path:	Parameter ...	MIME Type:

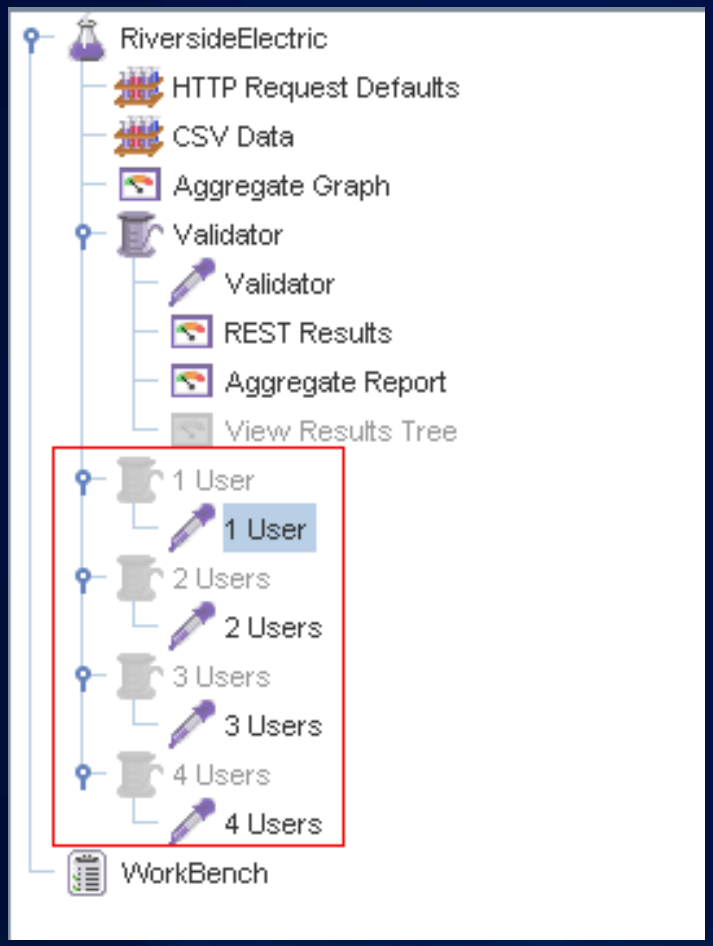
Add Browse... Delete

Optional Tasks

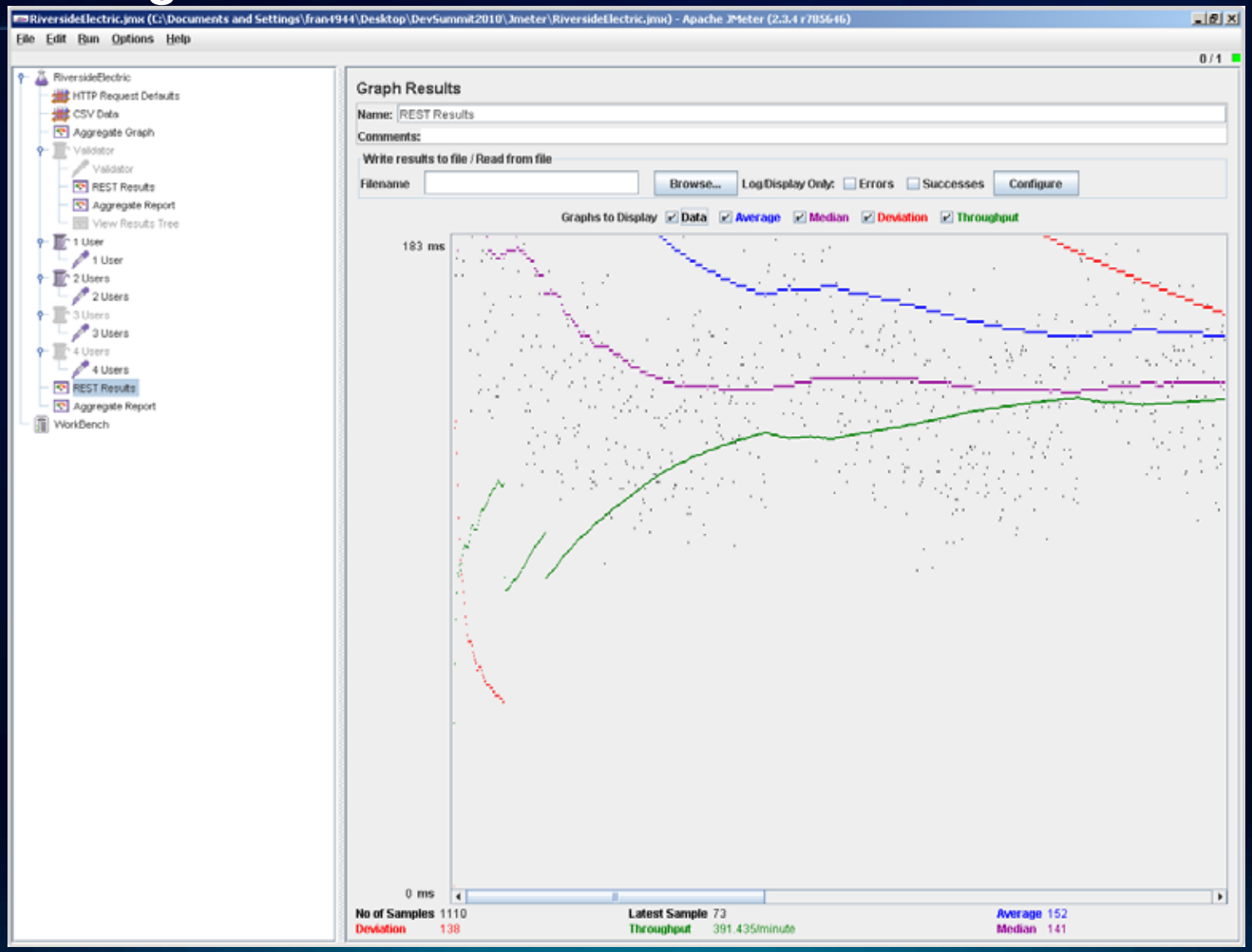
☐ Retrieve All Embedded Resources from HTML Files ☐ Use as Monitor ☐ Save response as MD5 hash?

Embedded URLs must match:

Testing with JMeter – Create Step Load



Testing with JMeter - Results



Testing with JMeter - Results

Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename | Browse... | Log/Display Only: Errors Successes | Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
1 User	1110	152	141	191	73	3289	0.09%	6.5/sec	136.2
TOTAL	1110	152	141	191	73	3289	0.09%	6.5/sec	136.2

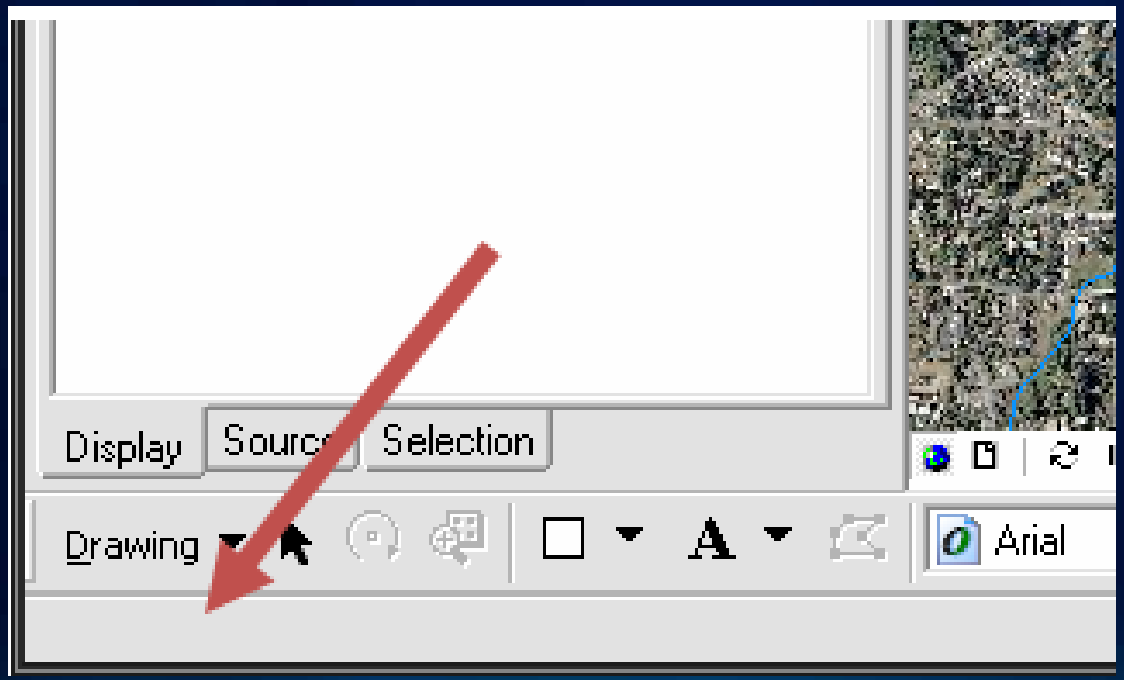
Desktop Testing

Performance Testing ArcMap and ArcEngine Applications

- Several tools can be used to automate thick client workflows
  - Citrix Edge Sight
  - HP LoadRunner
  - Borland’s Silk Perfomer
  - NRG’s AppLoader

Capturing Transactions in ArcMap Using Events

Better choices are text indicators that eventually have a definitive ending (such as text stating “complete”, or no text at all)



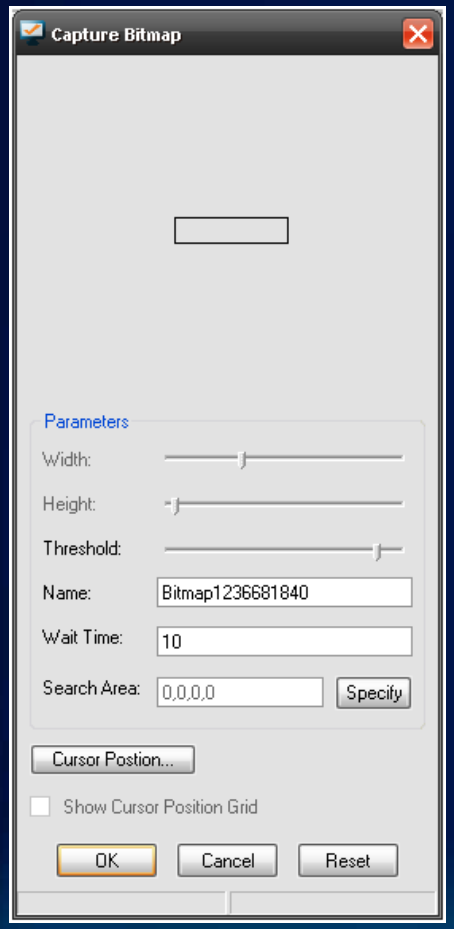
Capturing Transactions in ArcMap Using Events

Measuring pan and zoom transactions

- **WaitBitmapAppear:**
  - **lower left status text continuously update when drawing occurs, and will become blank when ended**
  - **Testing software waits until the bitmap is matched before proceeding.**
  - **Specify a search area when capturing the bitmap.**
  - **Keep a search area small**
  - **Animated icons or other moving status indicators are particularly difficult to implement (typical example is the spinning globe in ArcMap)**
  - **Better choices are text indicators that eventually have a definitive ending (such as text stating “complete”, or no text at all)**

Capturing Transactions in ArcMap Using Events

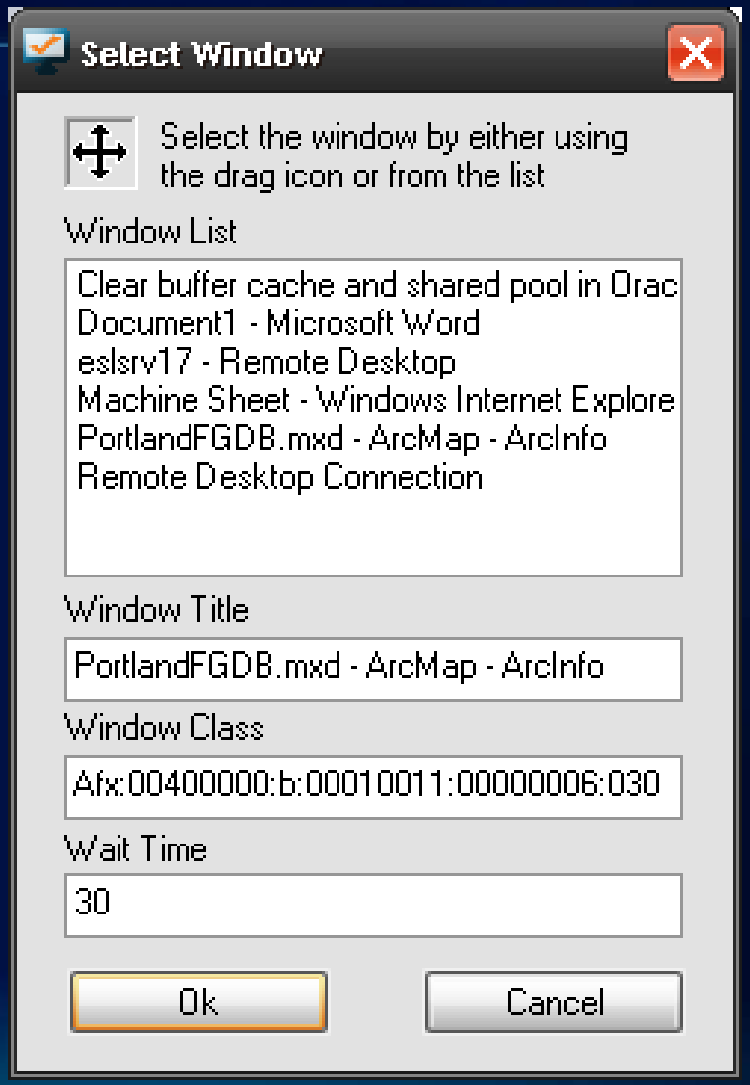
Measuring pan and zoom transactions (Apploader tool example)



Capturing Transactions in ArcMap Using Events

Measure opening ArcMap document (Apploader tool example)

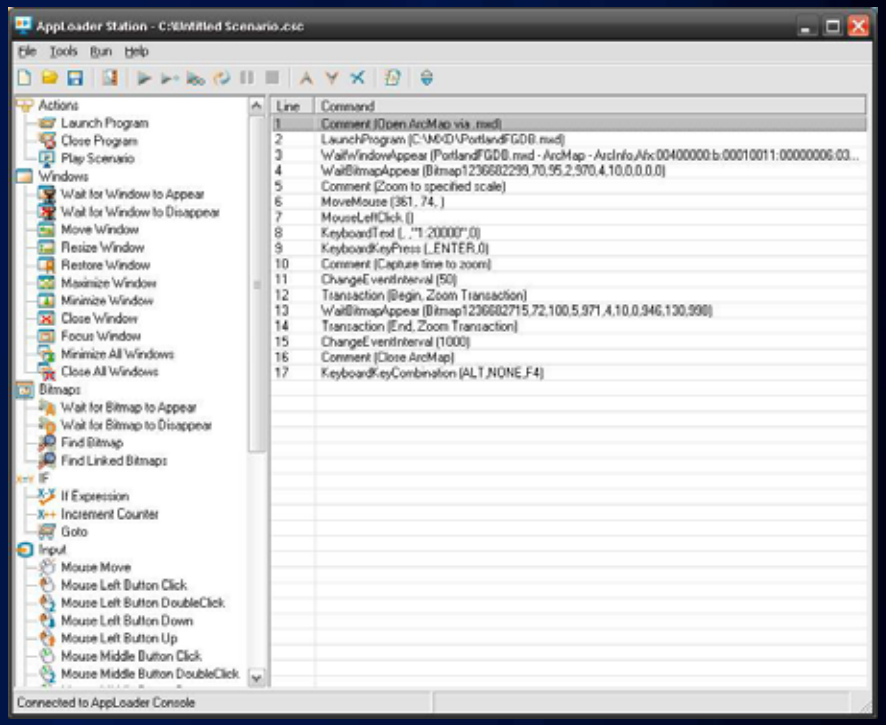
- **Wait for Window**



Capturing Transactions in ArcMap Using Events

Open ArcMap, Zoom, Close ArcMap

1. Comment : Open ArcMap
2. Launching ArcMap via specified MXD
3. Waiting for the ArcMap window to appear
4. Waiting for captured bitmap to appear. This bitmap is a blank status area indicating all drawing has completed.
5. Comment: Zoom
6. Move the mouse to the map scale field
7. Click the map scale field
8. Type the scale we wish to zoom to
9. Hit enter, initiating the map zoom
10. Comment: Start Zoom Transaction
11. Changing the Event Interval to 50ms for better



ArcGIS ADF Testing

Using Visual Studio for web testing ADF

- Creating web test script
  1. VS Recorder
     - might not detect all parameters for ADF
     - must validate with Fiddler or other tools
     - may have to edit registry entries
       - http://msdn.microsoft.com/en-us/library/cc678655.aspx
  2. Fiddler and export to VS Web test
     - Deselect appropriate plug-in
  3. Automated solutions (especially useful for ADF testing), e.g. Fiddler custom plug-in
- Declarative vs. Code web test
  - Looping and if logic can be implemented only in coded web test

Scripting ADF applications - Challenges

- Stateful application
  - Test scripts must manage the state
  - Most requests contain session based or dynamic GUID
    - Map Images
    - Scale bar
    - North Arrow
    - Search task
- Parameterize request

Creating Transaction Scripts – manual approach

- Most load testing tools include a an HTTP debugging proxy to model user workflows. The output from the HTTP Debugging proxy is raw text or another format that represents the Web Client requests.
- When recording user workflows, use transaction markers between the transactions. For Example, initial page load, zoom, and pan are transactions
- The transaction marker is also the correct place to insert think time as a user would naturally pause after a transaction

Creating Transaction Scripts - manual approach

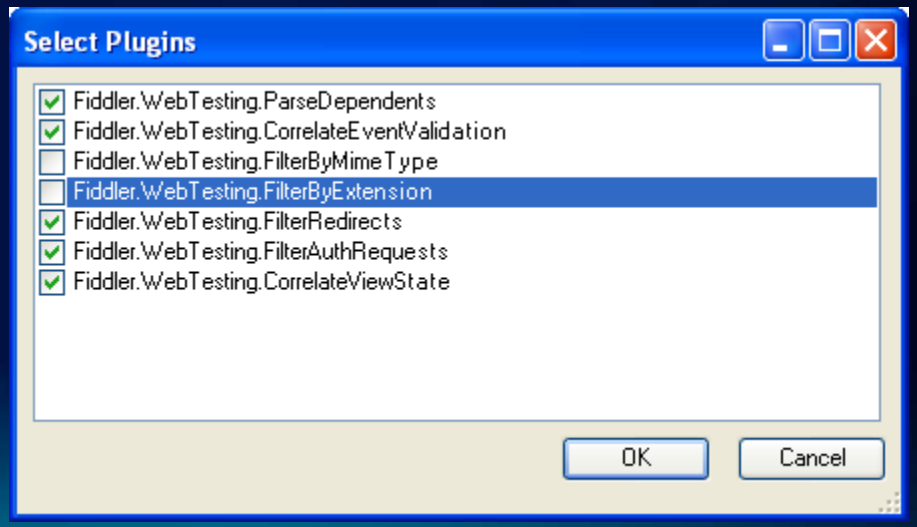
84	200	HTTP	eslsrv11	/TemplateMixedAggregate/WebResource.axd?d=InwIAjzTW5r2YzMrDnvYp-t8uIToH6LbDqCa-F
85	200	HTTP	eslsrv11	/TemplateMixedAggregate/WebResource.axd?d=InwIAjzTW5r2YzMrDnvYp-t8uIToH6LbDqCa-F
86	200	HTTP	eslsrv11	/TemplateMixedAggregate/WebResource.axd?d=InwIAjzTW5r2YzMrDnvYp-t8uIToH6LbDqCa-F
87	200	HTTP	eslsrv...	/TemplateMixedAggregate/ESRI.ArcGIS.ADF.Web.MimeImage.ashx?ImgID=d20
88	200	HTTP	eslsrv11	/TemplateMixedAggregate/WebResource.axd?d=InwIAjzTW5r2YzMrDnvYp-t8uIToH6LbDqCa-F
89	200	HTTP	eslsrv...	/TemplateMixedAggregate/ESRI.ArcGIS.ADF.Web.MimeImage.ashx?ImgID=c58
90	200	HTTP	eslsrv11	/TemplateMixedAggregate/images/dismiss.png
91	200	HTTP	eslsrv...	/TemplateMixedAggregate/ESRI.ArcGIS.ADF.Web.MimeImage.ashx?ImgID=5d5
92	502	HTTP	INI	/
93	200	HTTP	eslsrv11	/arcgiscache/PortlandCacheNoTaxLots/Portland/_alllayers/L06/R0000bda3/C0000a0e5.jpg
94	200	HTTP	eslsrv11	/arcgiscache/PortlandCacheNoTaxLots/Portland/_alllayers/L06/R0000bda4/C0000a0e5.jpg
95	200	HTTP	eslsrv11	/TemplateMixedAggregate/ESRI.ArcGIS.ADF.Web.UI.WebControls.MapHandler.ashx?MapID=
96	200	HTTP	eslsrv11	/TemplateMixedAggregate/Default.aspx
97	200	HTTP	eslsrv11	/arcgiscache/PortlandCacheNoTaxLots/Portland/_alllayers/L06/R0000bda3/C0000a0e6.jpg
98	200	HTTP	eslsrv11	/arcgiscache/PortlandCacheNoTaxLots/Portland/_alllayers/L06/R0000bda4/C0000a0e6.jpg
99	200	HTTP	eslsrv11	/arcgiscache/PortlandCacheNoTaxLots/Portland/_alllayers/L06/R0000bda3/C0000a0e7.jpg
100	200	HTTP	eslsrv11	/arcgiscache/PortlandCacheNoTaxLots/Portland/_alllayers/L06/R0000bda4/C0000a0e7.jpg
101	200	HTTP	eslsrv11	/TemplateMixedAggregate/images/backward.png
102	200	HTTP	eslsrv...	/TemplateMixedAggregate/ESRI.ArcGIS.ADF.Web.MimeImage.ashx?ImgID=b2e
103	200	HTTP	eslsrv...	/TemplateMixedAggregate/ESRI.ArcGIS.ADF.Web.MimeImage.ashx?ImgID=305
104	502	HTTP	Z1	/
105	200	HTTP	eslsrv11	/arcgiscache/PortlandCacheNoTaxLots/Portland/_alllayers/L07/R0001602f/C00012acf.jpg
106	200	HTTP	eslsrv11	/arcgiscache/PortlandCacheNoTaxLots/Portland/_alllayers/L07/R00016030/C00012acf.jpg
107	200	HTTP	eslsrv11	/arcgiscache/PortlandCacheNoTaxLots/Portland/_alllayers/L07/R00016031/C00012acf.jpg
108	200	HTTP	eslsrv11	/arcgiscache/PortlandCacheNoTaxLots/Portland/_alllayers/L07/R0001602f/C00012ad0.jpg
109	200	HTTP	eslsrv11	/TemplateMixedAggregate/Default.aspx
110	200	HTTP	eslsrv11	/TemplateMixedAggregate/ESRI.ArcGIS.ADF.Web.UI.WebControls.MapHandler.ashx?MapID=
111	200	HTTP	eslsrv11	/arcgiscache/PortlandCacheNoTaxLots/Portland/_alllayers/L07/R00016030/C00012ad0.jpg
112	200	HTTP	eslsrv11	/arcgiscache/PortlandCacheNoTaxLots/Portland/_alllayers/L07/R00016031/C00012ad0.jpg
113	200	HTTP	eslsrv11	/arcgiscache/PortlandCacheNoTaxLots/Portland/_alllayers/L07/R0001602f/C00012ad1.jpg
114	200	HTTP	eslsrv11	/arcgiscache/PortlandCacheNoTaxLots/Portland/_alllayers/L07/R00016030/C00012ad1.jpg
115	200	HTTP	eslsrv11	/arcgiscache/PortlandCacheNoTaxLots/Portland/_alllayers/L07/R00016031/C00012ad1.jpg
116	200	HTTP	eslsrv...	/TemplateMixedAggregate/ESRI.ArcGIS.ADF.Web.MimeImage.ashx?ImgID=210
117	200	HTTP	eslsrv...	/TemplateMixedAggregate/ESRI.ArcGIS.ADF.Web.MimeImage.ashx?ImgID=c65
118	502	HTTP	Z2	/

Creating Transaction Scripts – manual approach

- Using the transaction markers we will define GIS transactions.
- The content returned from the initial page request will contain dynamic images that must be captured, formatted properly, and sent back to the server in the form of an HTTP GET.
- All references to HTTP GET with ESRI.ArcGIS.Web.MimeImage.ashx must be altered to reflect the new session
- Transaction markers must be removed

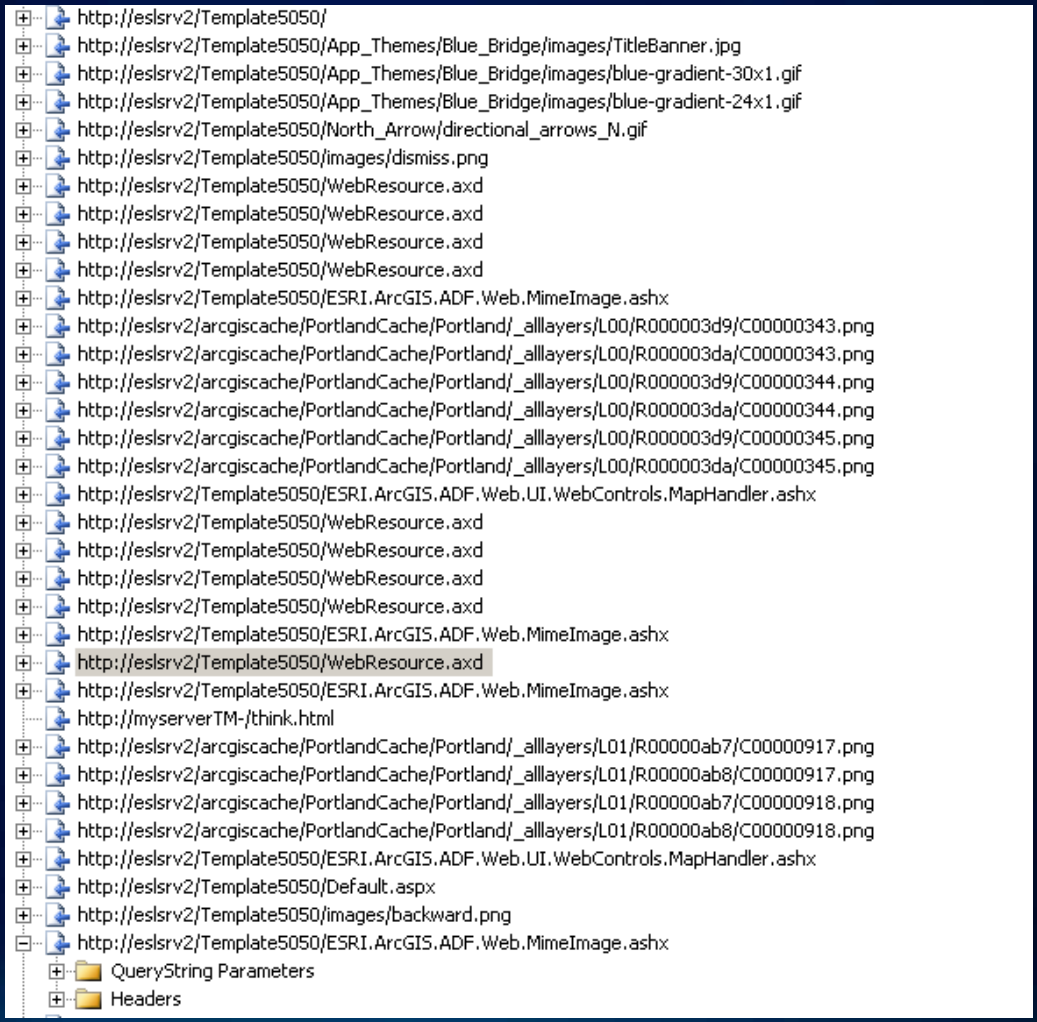
Creating Transaction Scripts – Fiddler manual

- Using Fiddler, an HTTP Debugging proxy, record the desired Web client workflow
- Be sure to include the transaction markers in your workflow.
- When you have completed recording the defined workflow, select all (Ctrl-A) and go to File->Save->Session(s)->as Visual Studio Web Test and save the raw HTTP traffic to fiddler.webtest. You will be prompted to select VS2008 plugins on export. Please select the following options-



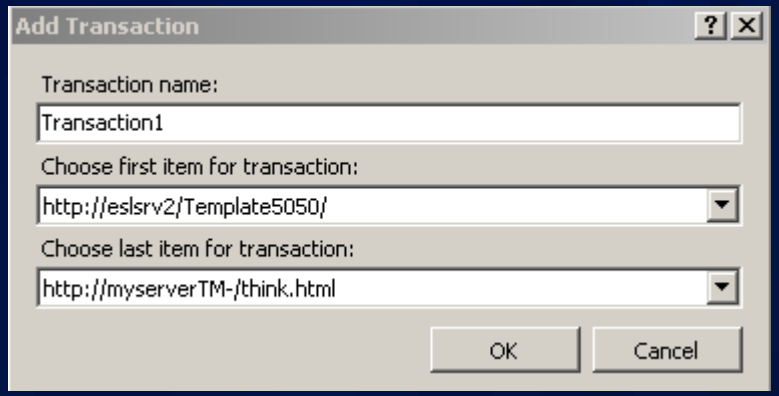
Creating Transaction Scripts – VS Web Test

- **Declarative webtest**



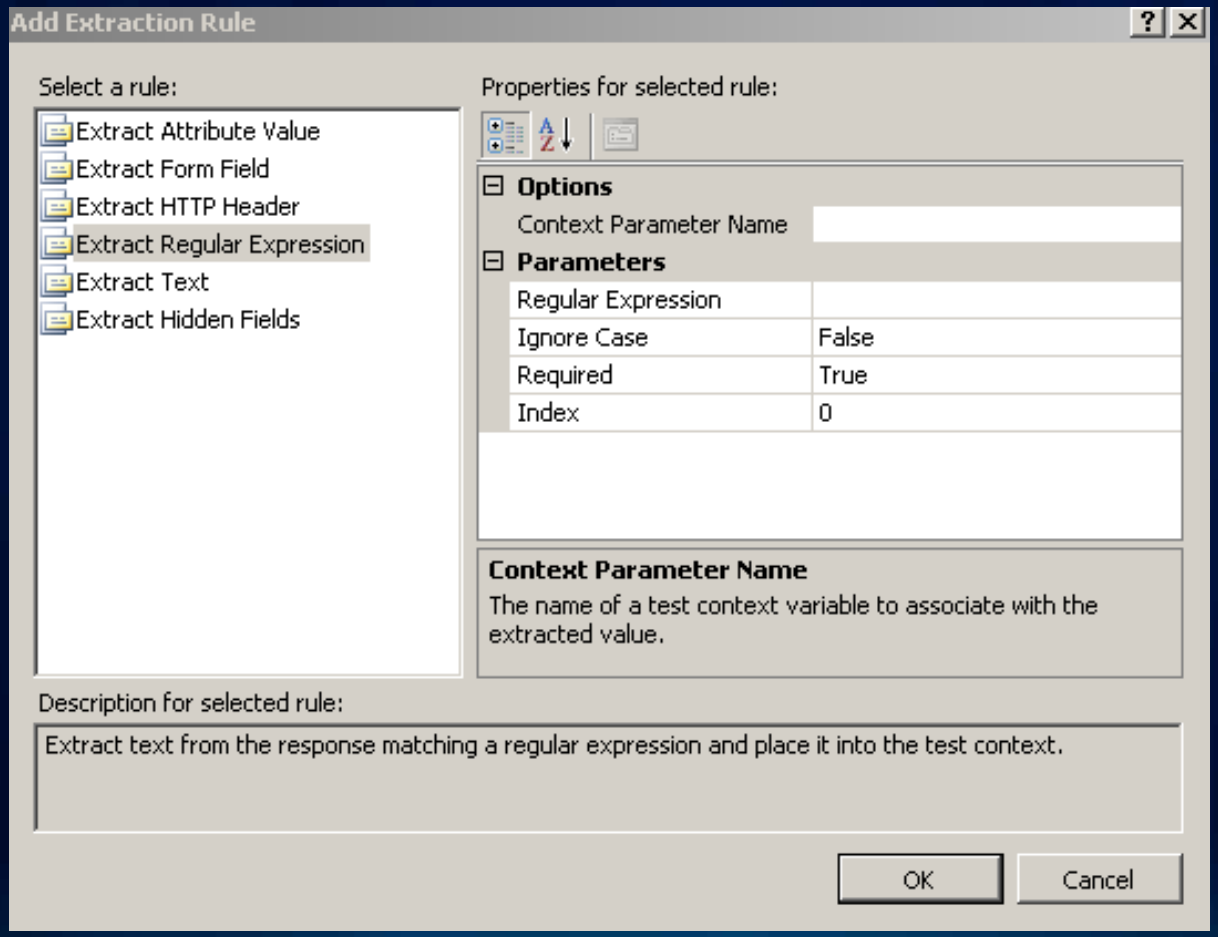
Creating Transaction Scripts – VS Web Test manual

- Group requests into transactions



- Define extraction rules for each web request that returns dynamic images.
  - Extraction rules define search patterns used to parse web requests, and the results of the search are held in a variable defined in extraction rule.

Creating Transaction Scripts – VS Web Test manual extraction Rules



Creating Transaction Scripts – VS Web Test manual extraction rules for ADF testing

Parameters	Value	Notes
Context Parameter Name	t0-i0	This item represents a variable that will be set when the extraction rule is executed on the response of an HTTP request. I recommend following the convention to the left. T0 represents transaction0 and i0 represent image0. If there is more than one request coming back in the response of the server or the define transaction has multiple dynamic images the Context Parameter name will simply be advanced. For Example, t0-i2.
Regular Expression	ImgID=((([0-9a-fA-F]){8}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){12})|([0-9a-fA-F]){32})	Dynamic images will have this signature, and the regular expression will find multiple images if they are present in the HTTP response from the server
Ignore Case	False	
Required	True	
Index	0	The regular expression will return an array of item that matches the search criteria. This number represents the index of the array that must be dereferenced.

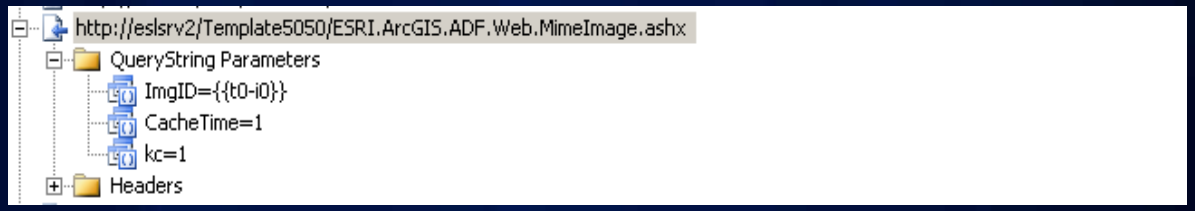
Creating Transaction Scripts – VS Web Test manual extraction rules for ADF testing

- Now, we have defined all of the extraction rules for the first transaction.
- Next, we must find the static requests from the initial HTTP Debugging proxy and modify them to be dynamic.
- Below is an example of a static request for a mime image that must be modified.
- We will delete all of the query string parameters, but we must leave the headers alone.

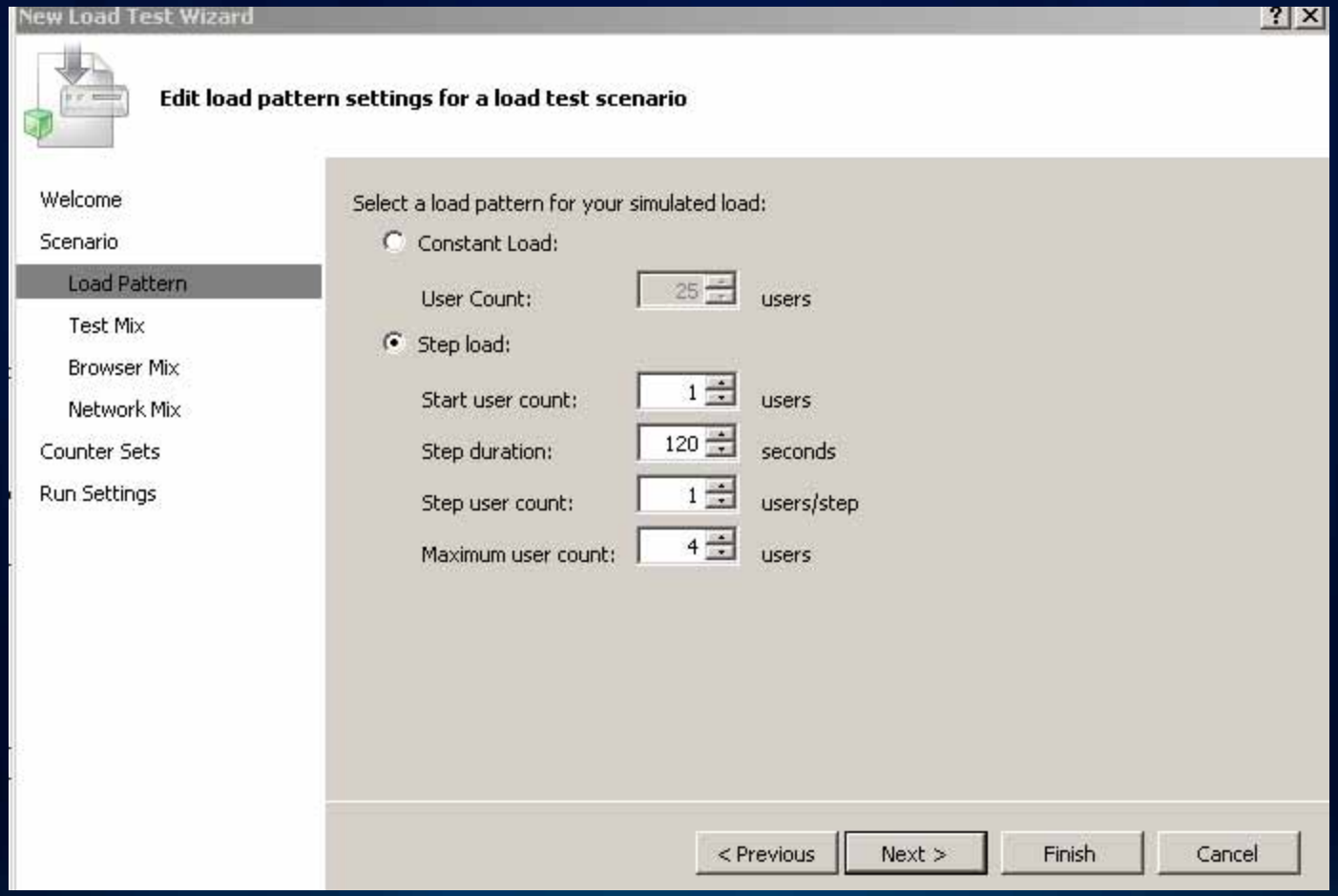
Creating Transaction Scripts – VS Web Test manual extraction rules for ADF testing

- **Now, we will use the variable defined by the Context Parameter Name in the extraction rules**
- **As you can see, we added the {{t0-i0}} to the correct request parameter. This modification to the request will dereference the variable that has been set when the extraction rule is executed.**

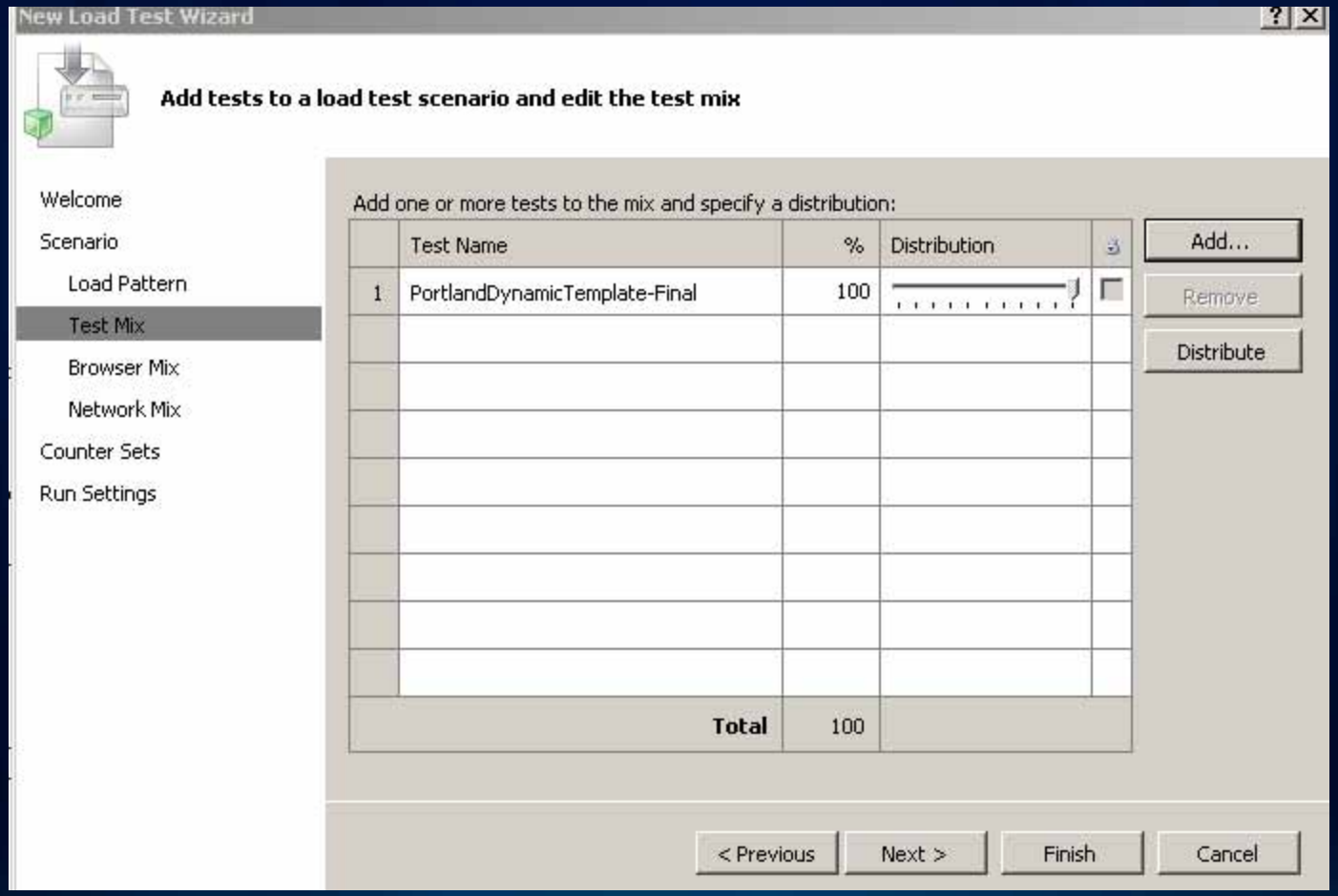
Modeling Web Clients



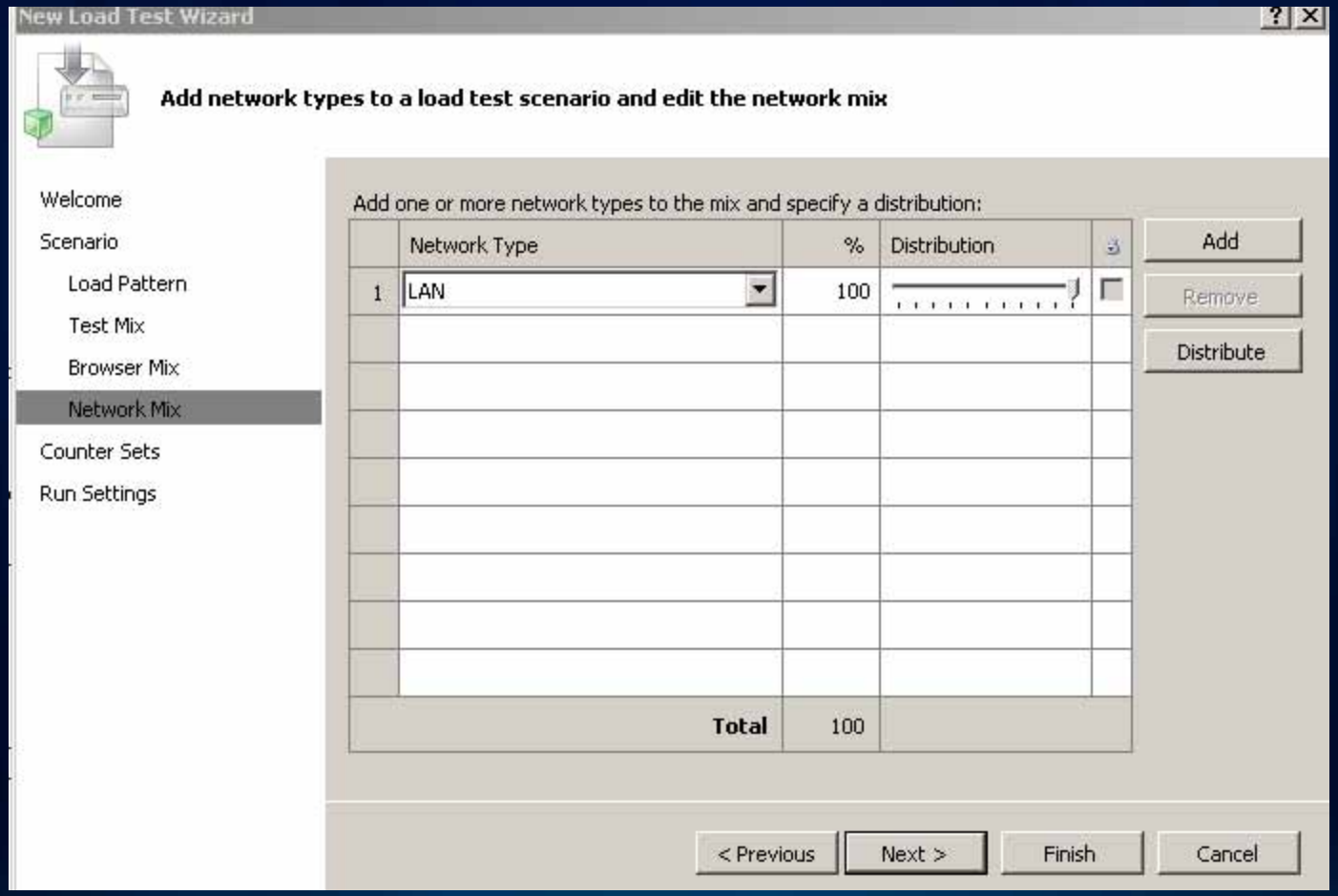
Creating load test – Visual Studio



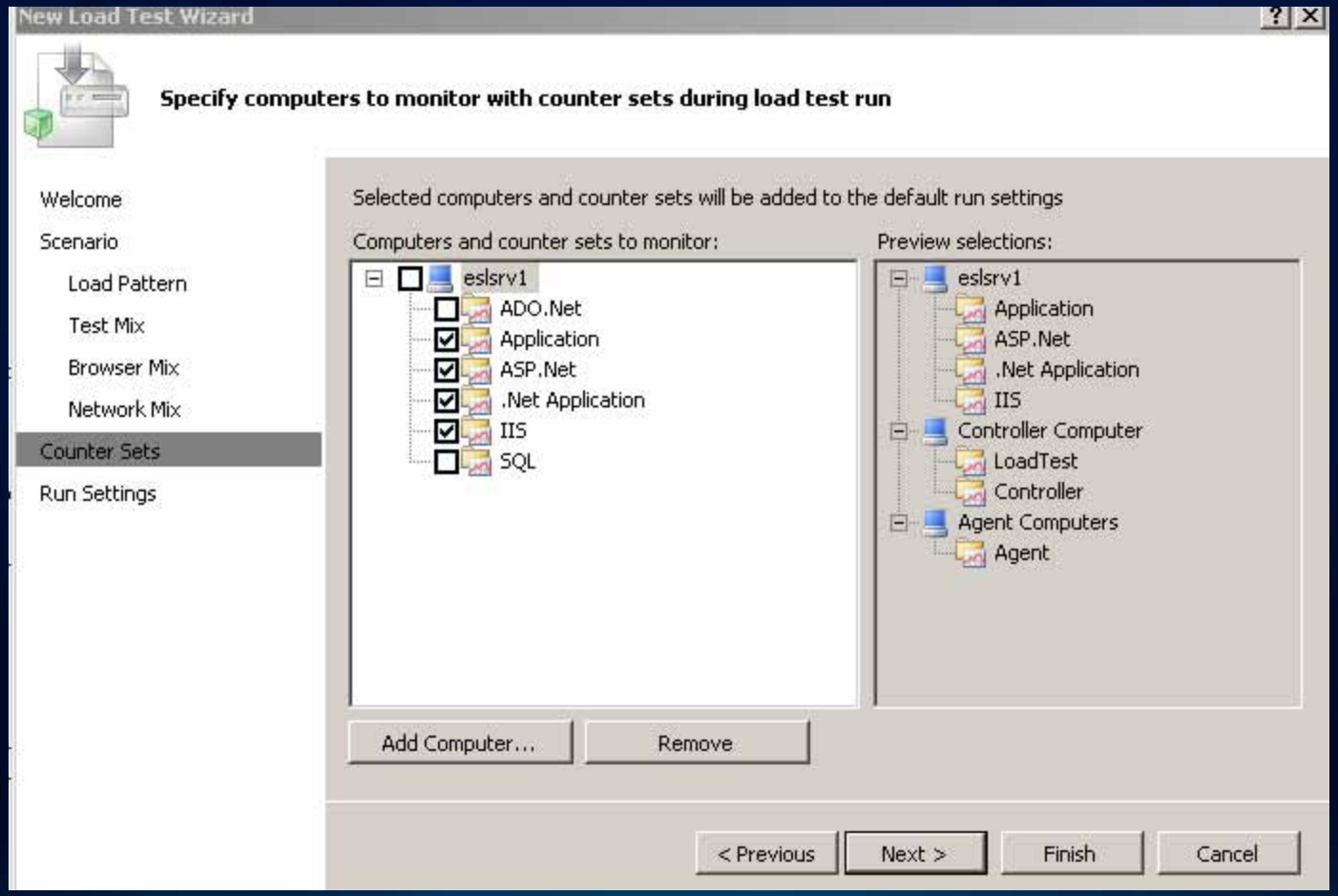
Creating load test – Visual Studio



Creating load test – Visual Studio



Creating load test – Visual Studio

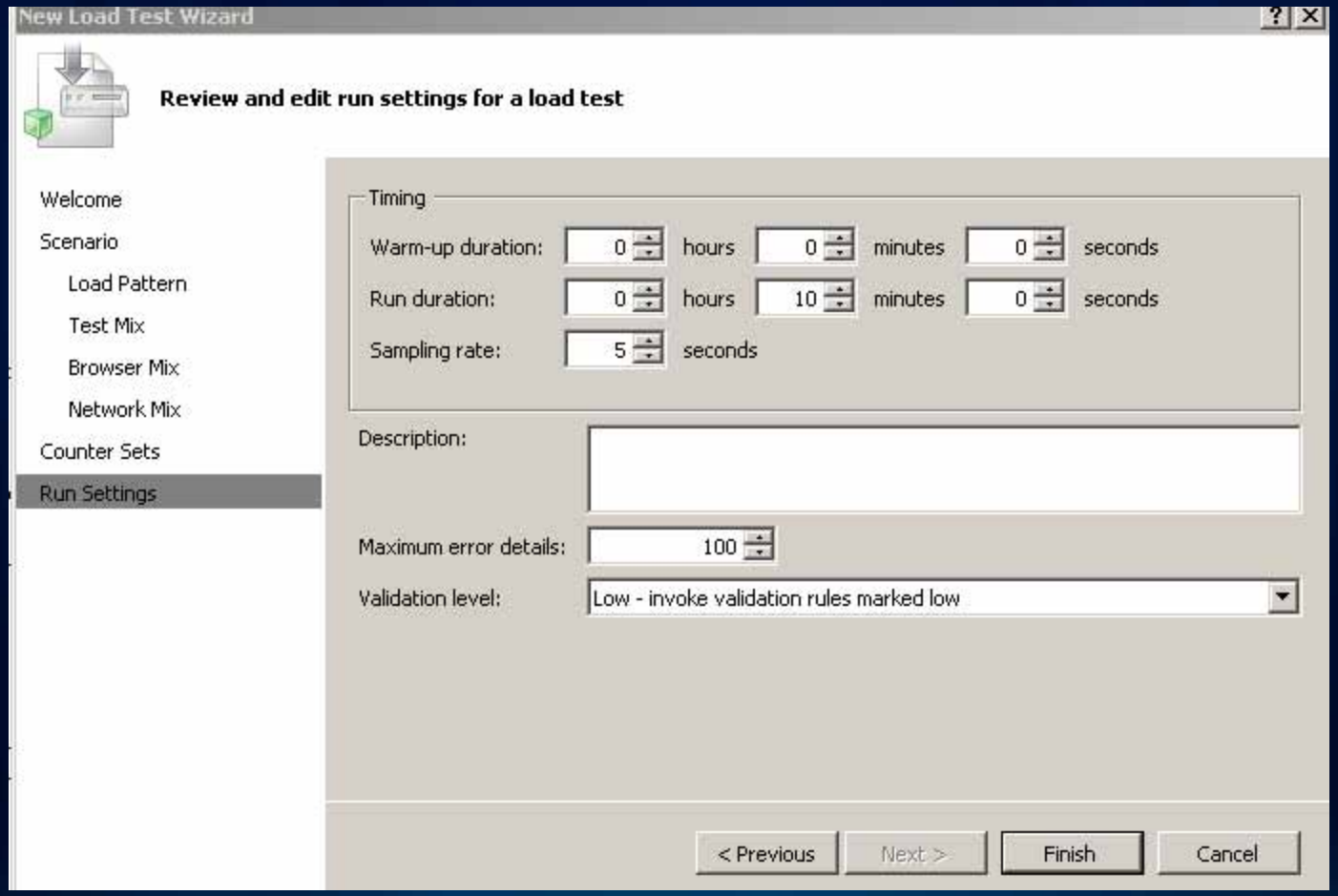


Running a Load Test with System Monitoring

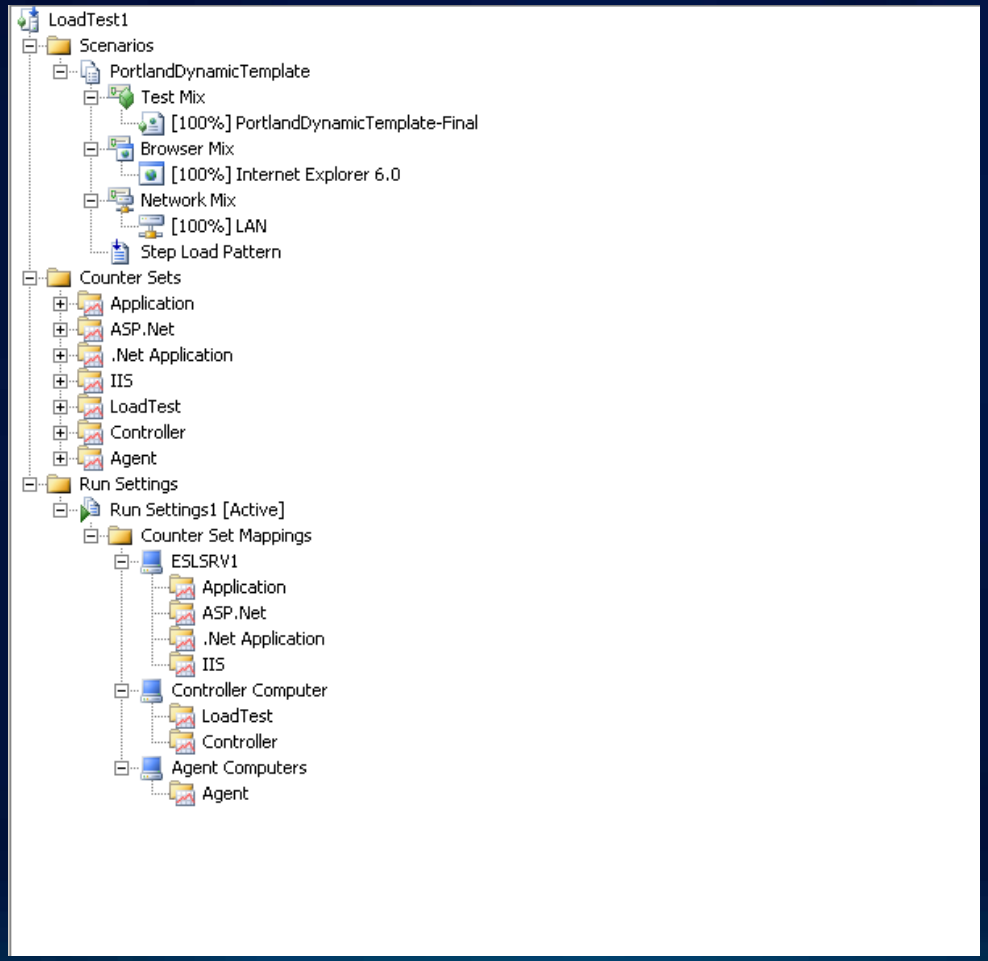
While the load test is running, various system components should be monitored including the CPUs, disk subsystem, memory, and network. System monitoring applies to all the systems that play a part in the test configuration, including the database server. At a minimum, the following core components should be monitored during the tests:

- CPU Utilization
- Memory
- Disk Utilization
- Network Utilization

Creating load test – Visual Studio



Creating load test – Visual Studio



Executing load test – Visual Studio