Vistronix, Inc®

Ingenuityinnovationintegrity

An Integral Part of your Technology
and Business Management Solutions
Managed Services
Information Management
Technical Solutions

Vistronix is a leader in Technology Solutions and Information Management Services.

2013 Esri International User Conference

July 8–12, 2013 | San Diego, California

Building Cache Tiles with ArcGIS Server 10.1
Real World Examples
from USDA/NRCS/NGCE

Presenter: Art Ullman
Contributor: Dr. Suling Zhao

DATE: 3/20/2013

1851 Alexander Bell Drive   Suite 350   Reston, VA 20191
703.734.2270   800.483.2434   www.vistronix.com

Project Overview

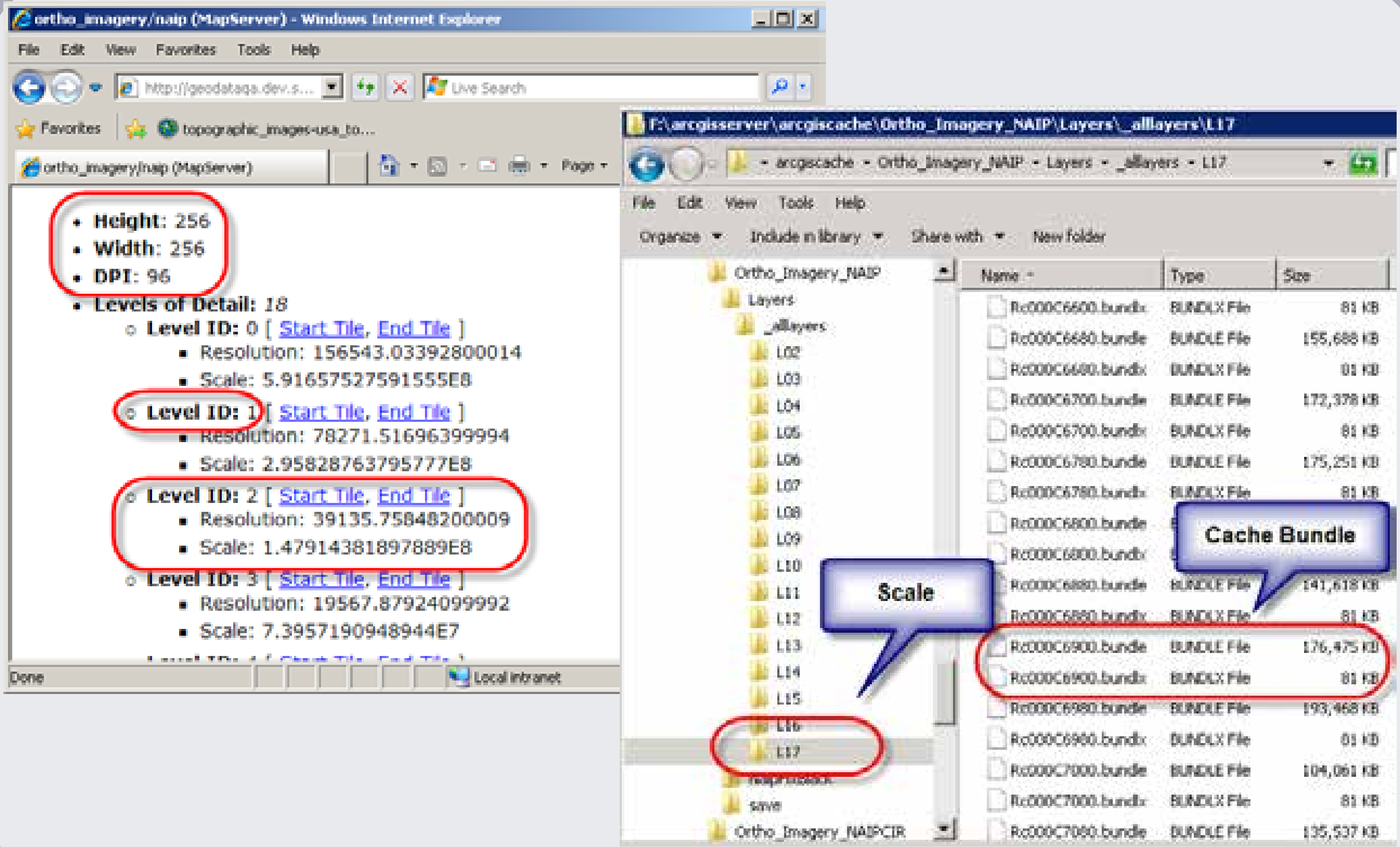
- USDA / NRCS has a requirement for high performance Geospatial Web Services to support planning and implementation of conservation practices for farmers and ranchers across all 50 states and territories.
- NRCS users need imagery, elevation data, and vector data delivered over USDA Network to web applications and ArcGIS Desktop.
- Support over 10,000 users in over 3000 counties and over 50TB of geospatial data and imagery.
- National Geospatial Center of Excellence (NGCE) at NRCS is responsible for providing the data and building web services.

Project Overview

Solution –

- Geospatial Web Service based on ArcGIS Server 10.1 SP1
- Use cache tile services to improve performance and scalability for imagery and complex vector layers.
- Vistronix and NGCE staff at the *National Geospatial Center of Excellence* in Fort Worth, Texas will build services using NRCS Geospatial Data Warehouse data and hardware.
- Use ESRI appliance cache tiles for topo maps, transportation and other layers available on the appliance.

Cache Tile Services - Overview



ortho_imagery/naip (MapServer) - Windows Internet Explorer
File Edit View Favorites Tools Help
http://geodataqa.dev.s...
Live Search
Favorites
topographic_images-usa_to...
ortho_imagery/naip (MapServer)
Page
Done
Local intranet

- **Height:** 256
- **Width:** 256
- **DPI:** 96
- **Levels of Detail:** *18*
  - **Level ID:** 0 [Start Tile, End Tile]
    - Resolution: 156543.03392800014
    - Scale: 5.91657527591555E8
  - **Level ID:** 1 [Start Tile, End Tile]
    - Resolution: 78271.51696399994
    - Scale: 2.95828763795777E8
  - **Level ID:** 2 [Start Tile, End Tile]
    - Resolution: 39135.75848200009
    - Scale: 1.47914381897889E8
  - **Level ID:** 3 [Start Tile, End Tile]
    - Resolution: 19567.87924099992
    - Scale: 7.3957190948944E7
  - **Level ID:** 4 [Start Tile, End Tile]

F:\arcgisserver\arcgiscache\Ortho_Imagery_NAIP\Layers_alllayers\L17
arcgiscache ▸ Ortho_Imagery_NAIP ▸ Layers ▸ _alllayers ▸ L17
File Edit View Tools Help
Organize ▾ Include in library ▾ Share with ▾ New folder
Ortho_Imagery_NAIP
Layers
_alllayers
L02
L03
L04
L05
L06
L07
L08
L09
L10
L11
L12
L13
L14
L15
L16
L17
nearptrdblock
save
Ortho_Imagery_NAIPCIR

Name	Type	Size
R0c00C6600.bundlx	BUNDLX File	81 KB
R0c00C6680.bundle	BUNDLE File	155,688 KB
R0c00C6680.bundlx	BUNDLX File	81 KB
R0c00C6700.bundle	BUNDLE File	172,378 KB
R0c00C6700.bundlx	BUNDLX File	81 KB
R0c00C6780.bundle	BUNDLE File	175,251 KB
R0c00C6780.bundlx	BUNDLX File	81 KB
R0c00C6800.bundle		
R0c00C6800.bundlx		
R0c00C6880.bundle	BUNDLE File	141,618 KB
R0c00C6880.bundlx	BUNDLX File	81 KB
R0c00C6900.bundle	BUNDLE File	176,475 KB
R0c00C6900.bundlx	BUNDLX File	81 KB
R0c00C6980.bundle	BUNDLE File	193,468 KB
R0c00C6980.bundlx	BUNDLX File	81 KB
R0c00C7000.bundle	BUNDLE File	104,061 KB
R0c00C7000.bundlx	BUNDLX File	81 KB
R0c00C7080.bundle	BUNDLE File	135,537 KB

Cache Bundle

Scale

Vistronix, Inc®

Cache Tile Services - Overview

- Why use cache tile services?
  - Performance
    - Eliminate complex queries and on-the-fly image/mosaic rendering on server.
  - Scalability
    - Server simply reads and sends JPG/PNGs to client. Scalability is determined by how fast the server can read and send JPG/PNGs to client.
  - User Experience
    - Users get continual flow of tiles as they pan and zoom.
    - Application is always responsive and continuously gives visual feedback to the user as tiles are delivered.
    - Users are used to this kind of experience from using Google, ESRI Online and BING maps.

Cache Tile Services – Key Steps

q Design

- q Projection – Web Mercator Auxiliary Sphere in most cases.
- q Cartography - Symbology, labeling, etc.
- q Scales, extent and image format.
- q Tile Scheme – ESRI/Google/BING

q Develop

- q Prepare input datasets
- q Hardware – non-production servers / workstations for building tiles. One large multi-processor machine or a cluster of lower power machines.
- q Storage – each scale requires 3-4 times storage of previous scale.
- q Time – product of spatial extent and scale divided by CPU count.
- q Test small areas and view results!!!

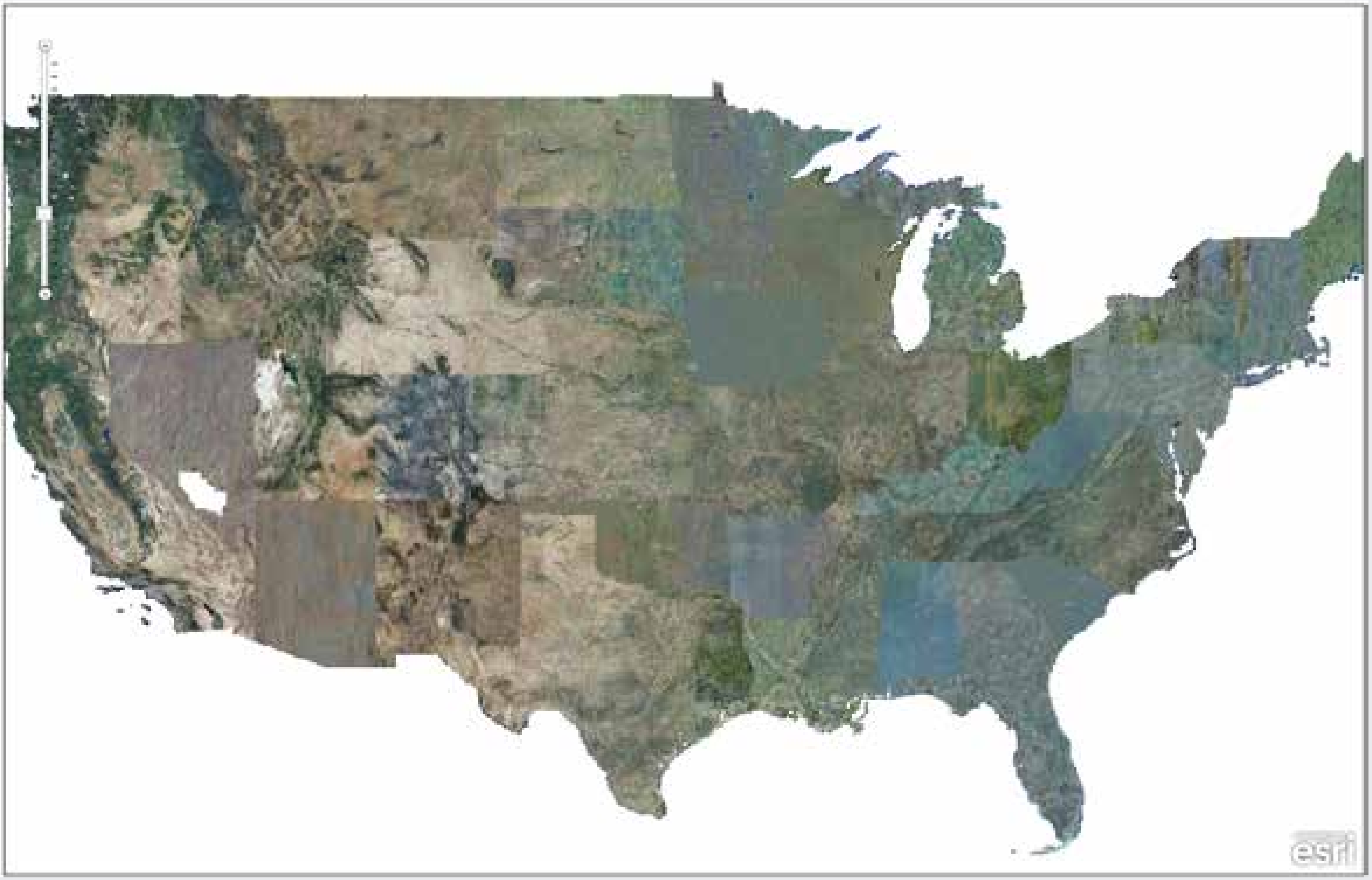
q Deploy

- q Deploy data and cache - or just cache?
- q Copy cache to production server and publish service.

NRCS / NGCE Cache Tile Services

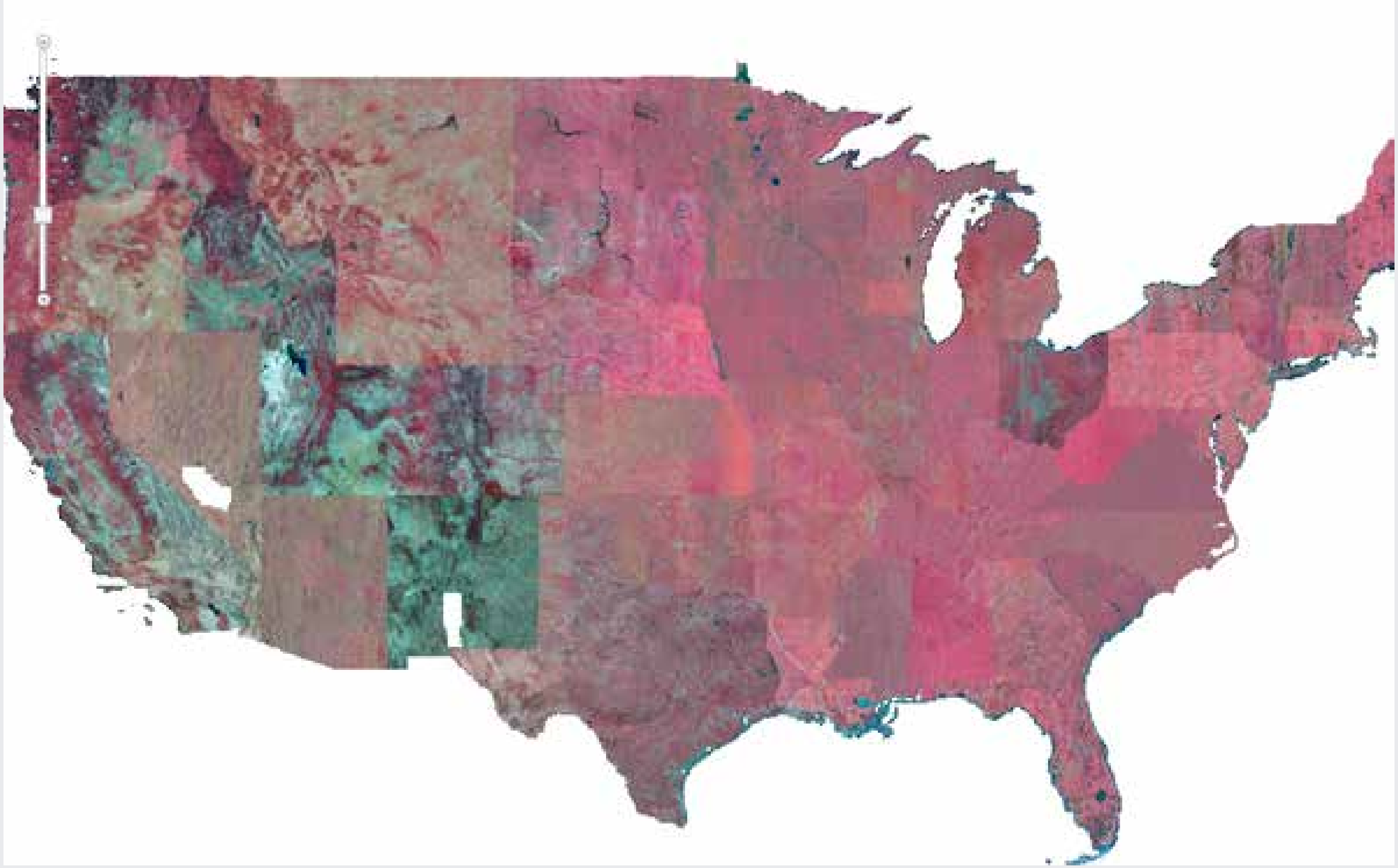
Layer	Type	Size	Max Scale / Resolution
NAIP – Natural Color	Imagery	2.15 TB	1:4513 -1.19m
NAIP – Color Infrared	Imagery	2.10 TB	1:4513 – 1.19m
NHD - Hydrography	Vector	516 GB	1:4513 – 1.19m
Relief - Elevation	Raster	112 GB	1:36111 – 9.55m
WBD – Hydrologic Units	Vector	93.8 GB	1:4513 – 1.19m
State / County	Vector	87.4 GB	1:9026 – 2.38m
Contours	Vector	689 GB	1:4513 – 1.19m
(5) Easement Imagery Layers	Imagery	620 GB each	1:1128 - 0.29m
(7) layers from ESRI Appliance including topo maps and transportation	Img & Vector	27 TB	mixed

NAIP 4 band 1m – Natural Color - bands: 1,2,3



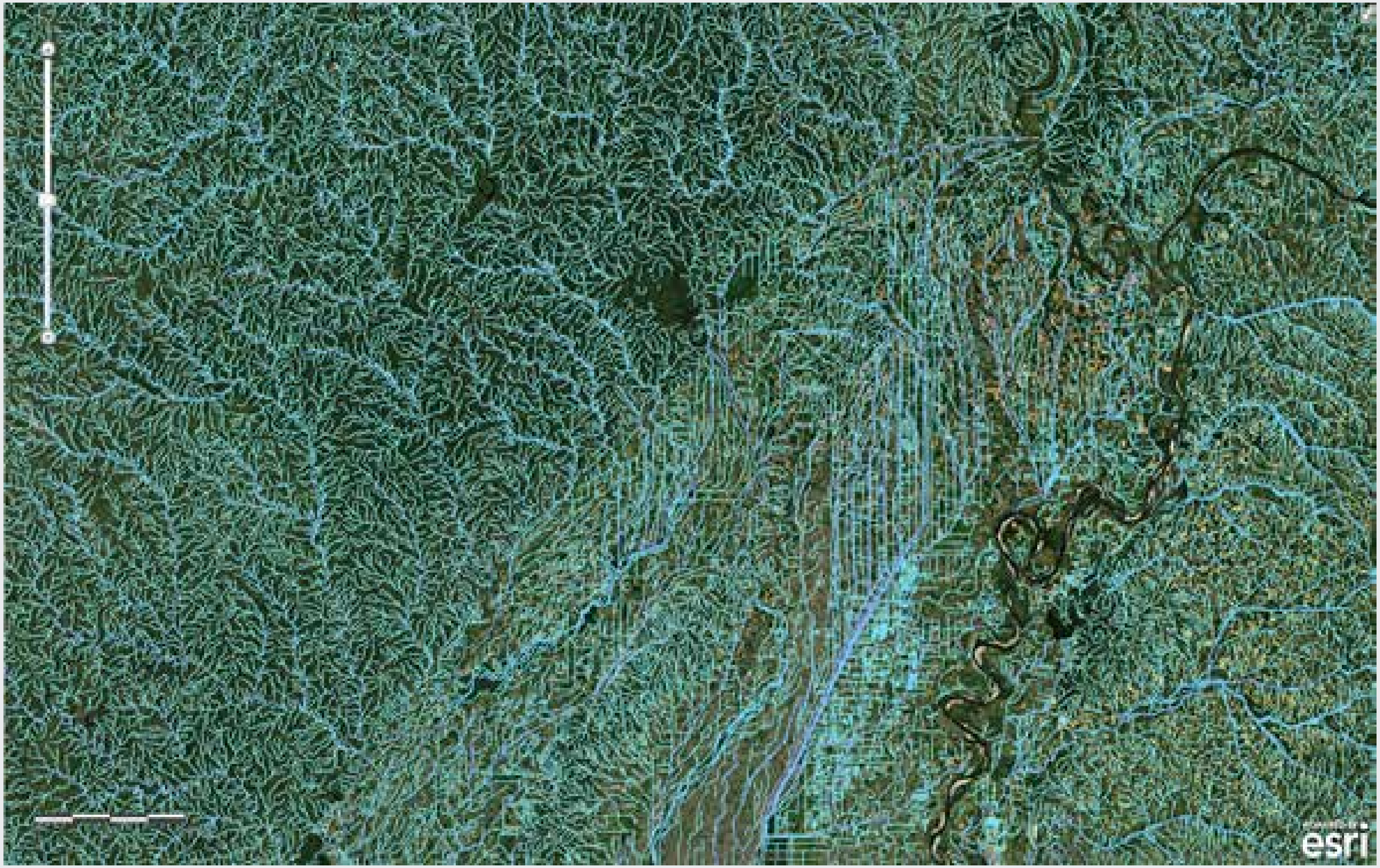
Vistronix, Inc®

NAIP 4 band 1m – Color Infra-red layer – Bands: 4,1,2

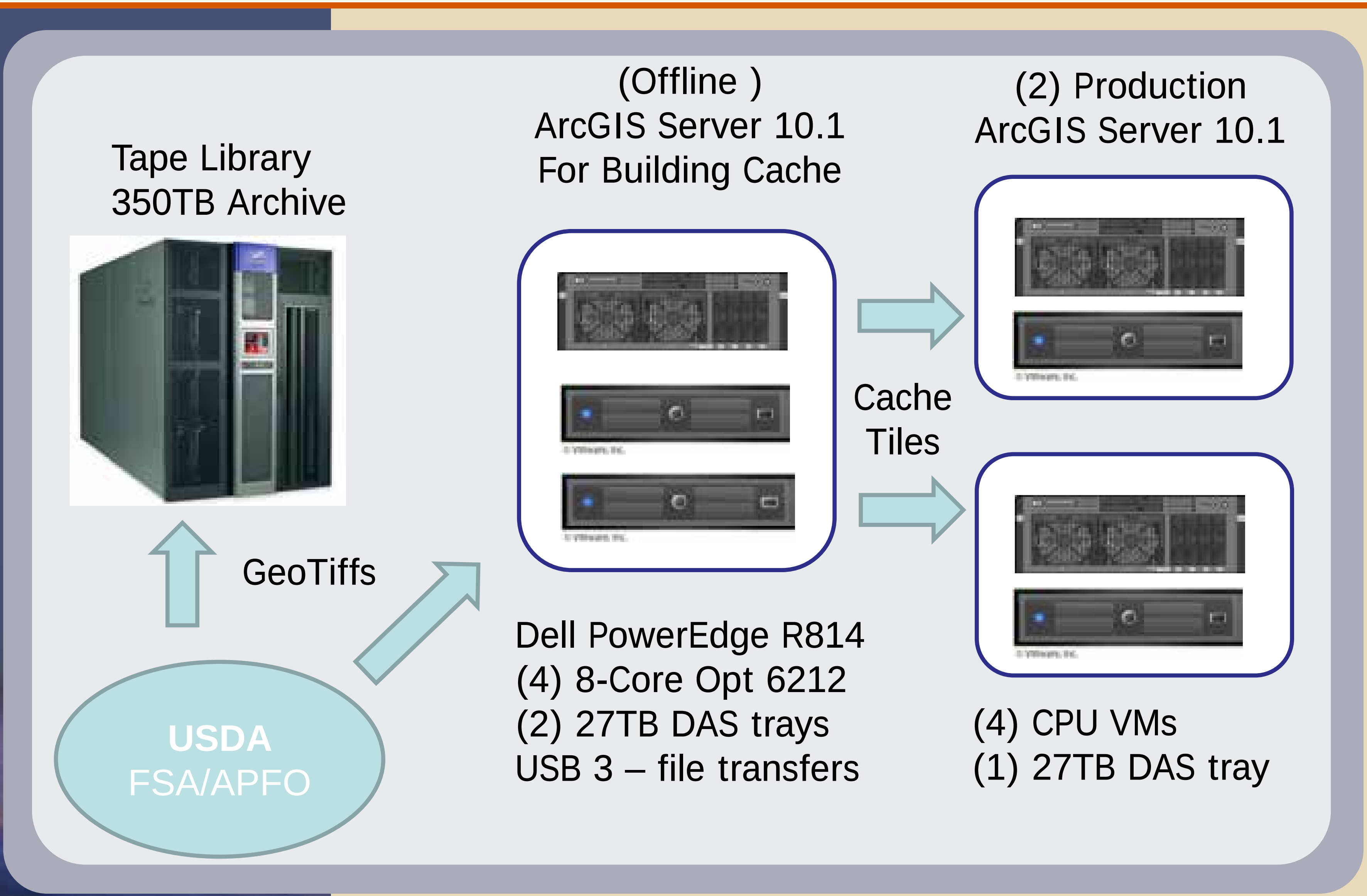


Vistronix, Inc®

Transparent (PNG) National Hydrography Dataset cache tile – draped over imagery tiles



Architecture – NAIP Imagery Processing



Tape Library
350TB Archive

GeoTiffs

USDA
FSA/APFO

(Offline)
ArcGIS Server 10.1
For Building Cache

Dell PowerEdge R814
(4) 8-Core Opt 6212
(2) 27TB DAS trays
USB 3 – file transfers

Cache
Tiles

(2) Production
ArcGIS Server 10.1

(4) CPU VMs
(1) 27TB DAS tray

Imagery – NAIP

§Input dataset overview

- Ø1m NAIP (National Agriculture Imagery Program)
- Ø4 band – produce two cache services: Natural Color (RGB) and Color Infra-Red (CIR)
- ØYearly refresh of 1/3 of states
- ØIncludes full coverage for lower 48 states.
- Ø210,000 geotiffs – 23.2 TB
  - ØPre-2012 – 20MB JPG compressed geotiffs (3.2TB)*
  - Ø2012 – 200MB uncompressed geotiffs (20TB)

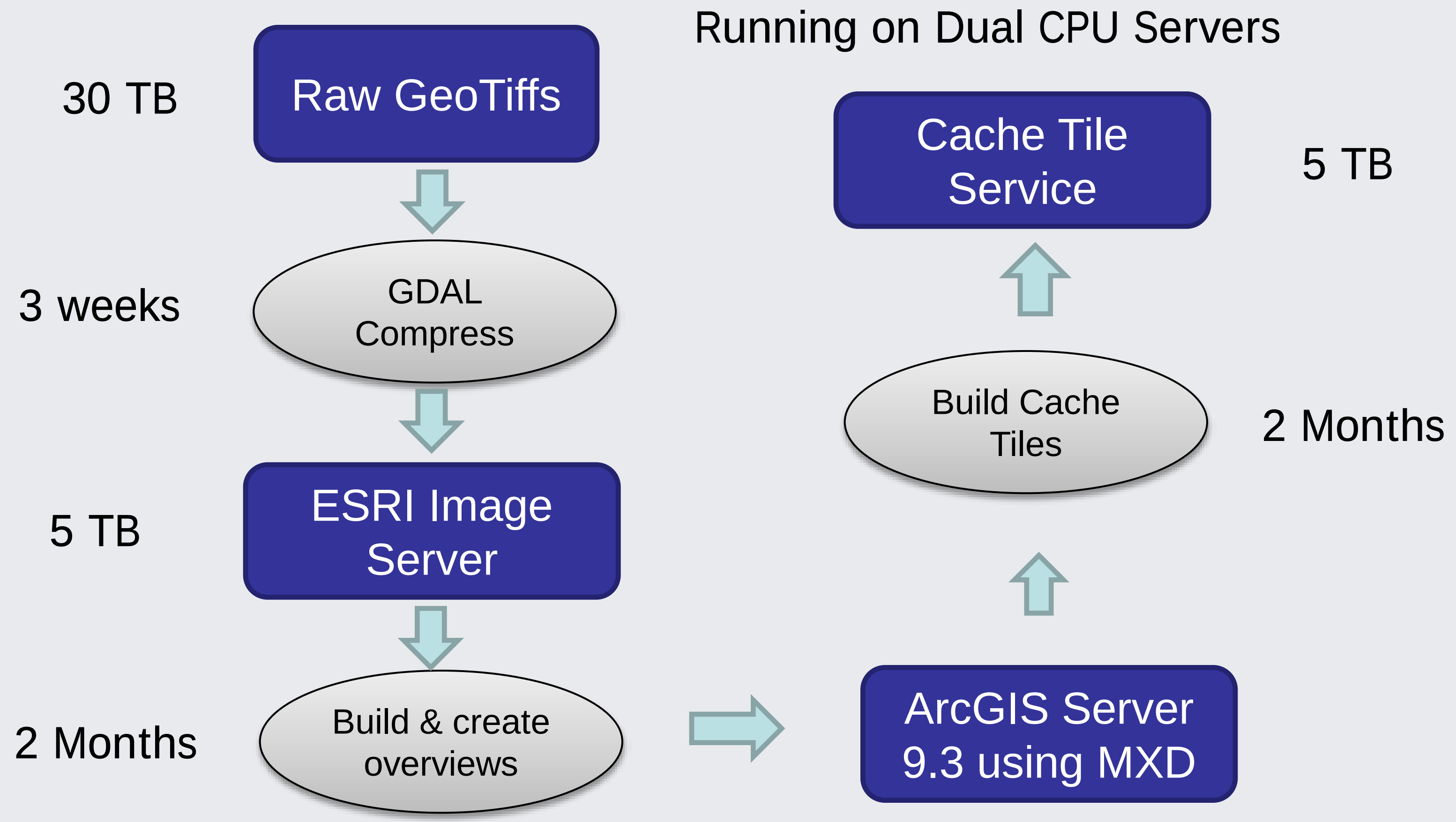
* Pre-2012 NAIP data was compressed and loaded to ESRI Image Server. Compression was needed to save space.

Imagery

§Design

- ØSource Data - 30TB 4band 1m resolution quarter-quads geotiffs
- ØMosaic Dataset for RGB – bands 1,2,3
- ØReference Mosaic Dataset for CIR – bands 4,1,2
- Ø2 Cache Tile Services (RGB/CIR)
- ØMixed Mode
  - ØJPEG for all internal tiles
  - ØPNG32 for all tiles that have empty areas - border areas.
- ØGoogle/BING/ESRI Tile Scheme
- ØScales: 1:145,000,000 – 1:4,513 (1.1 m)

Old 9.3 / 10.0 Processing Workflow



Running on Dual CPU Servers

30 TB — Raw GeoTiffs

3 weeks — GDAL Compress

5 TB — ESRI Image Server

2 Months — Build & create overviews

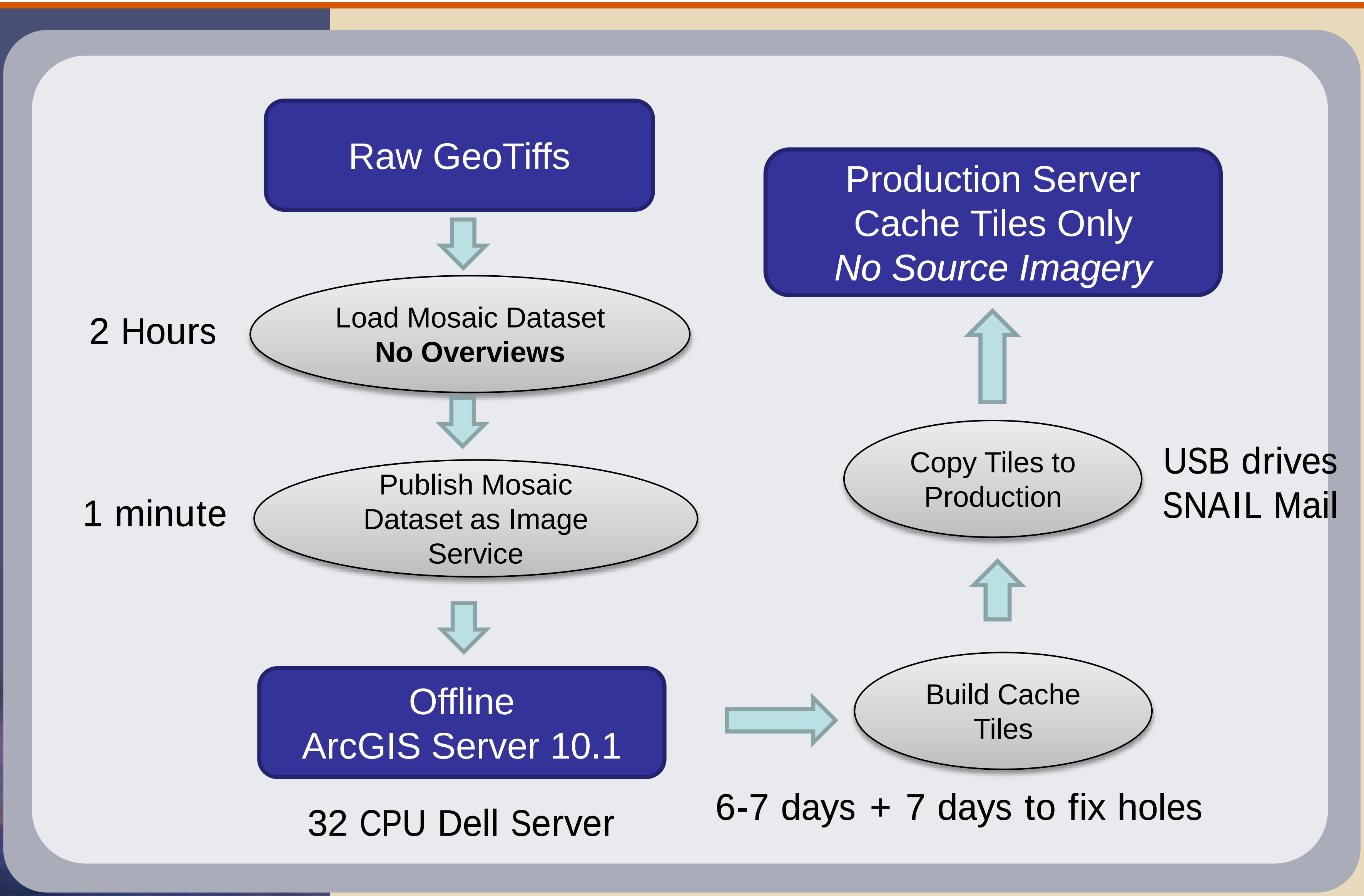
ArcGIS Server 9.3 using MXD

Build Cache Tiles — 2 Months

Cache Tile Service — 5 TB

Vistronix, Inc®

10.1 Processing Workflow



Raw GeoTiffs

2 Hours

Load Mosaic Dataset
No Overviews

1 minute

Publish Mosaic Dataset as Image Service

Offline
ArcGIS Server 10.1

32 CPU Dell Server

Build Cache Tiles

6-7 days + 7 days to fix holes

Copy Tiles to Production

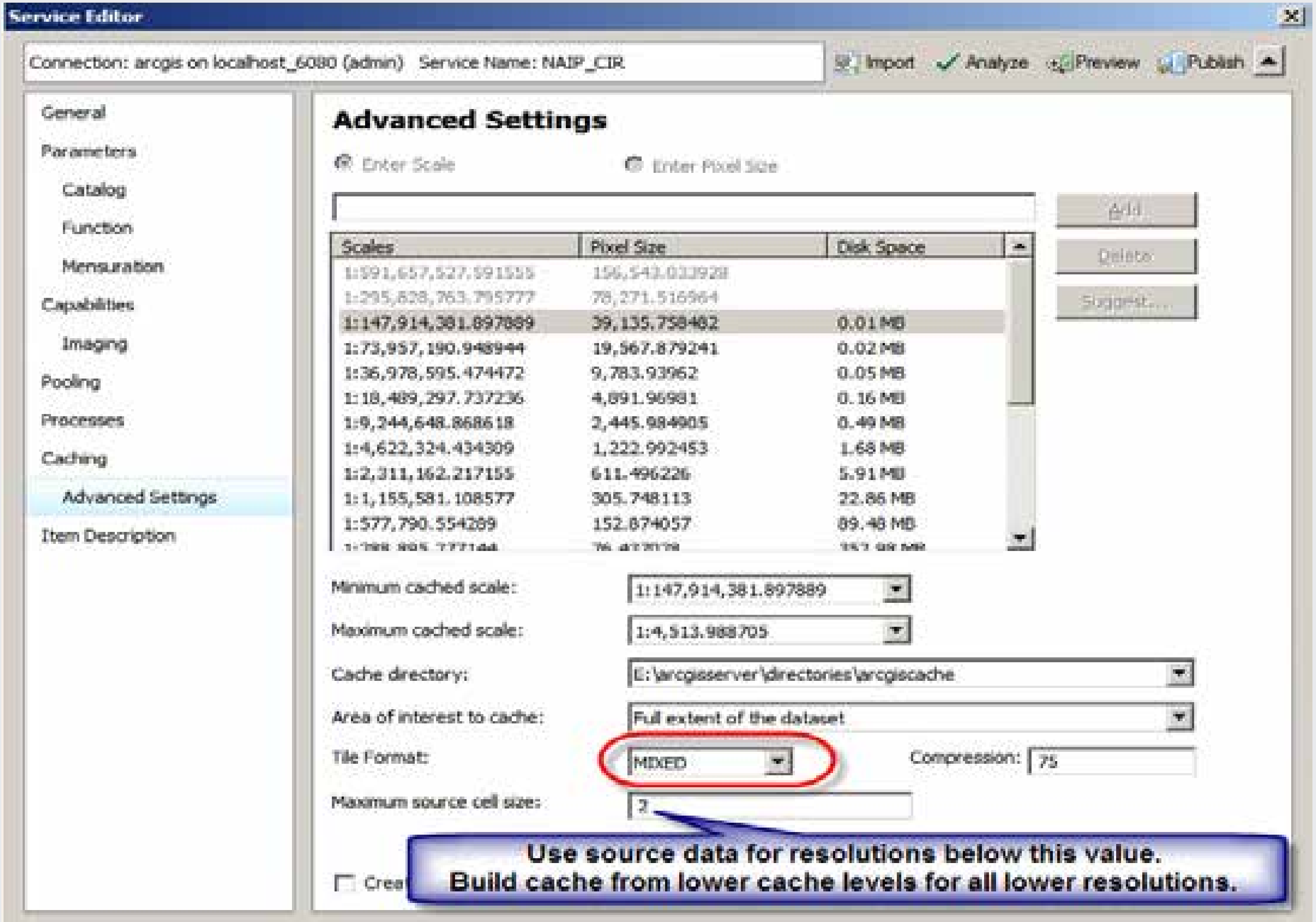
USB drives
SNAIL Mail

Production Server
Cache Tiles Only
No Source Imagery

Building Cache for Imagery

- Build Mosaic Dataset (MD)
- Add Rasters to MD
- Fix Boundary if necessary
- Calculate Statistics
- Publish Mosaic Dataset as an Image Service. Note: requires Image Server Extension for ArcGIS Server
- Set Caching Parameters
- Start process to build cache

Imagery – Building Cache



Connection: arcgis on localhost_6080 (admin)   Service Name: NAIP_CIR

Advanced Settings

● Enter Scale   ○ Enter Pixel Size

Scales	Pixel Size	Disk Space
1:591,657,527.591555	156,543.033928	
1:295,828,763.795777	78,271.516964	
1:147,914,381.897889	39,135.758482	0.01 MB
1:73,957,190.948944	19,567.879241	0.02 MB
1:36,978,595.474472	9,783.93962	0.05 MB
1:18,489,297.737236	4,891.96981	0.16 MB
1:9,244,648.868618	2,445.984905	0.49 MB
1:4,622,324.434309	1,222.992453	1.68 MB
1:2,311,162.217155	611.496226	5.91 MB
1:1,155,581.108577	305.748113	22.86 MB
1:577,790.554289	152.874057	89.48 MB
1:288,895.277144	76.437028	357.98 MB

Minimum cached scale: 1:147,914,381.897889

Maximum cached scale: 1:4,513.988705

Cache directory: E:\arcgisserver\directories\arcgiscache

Area of interest to cache: Full extent of the dataset

Tile Format: MIXED   Compression: 75

Maximum source cell size: 2

□ Creat

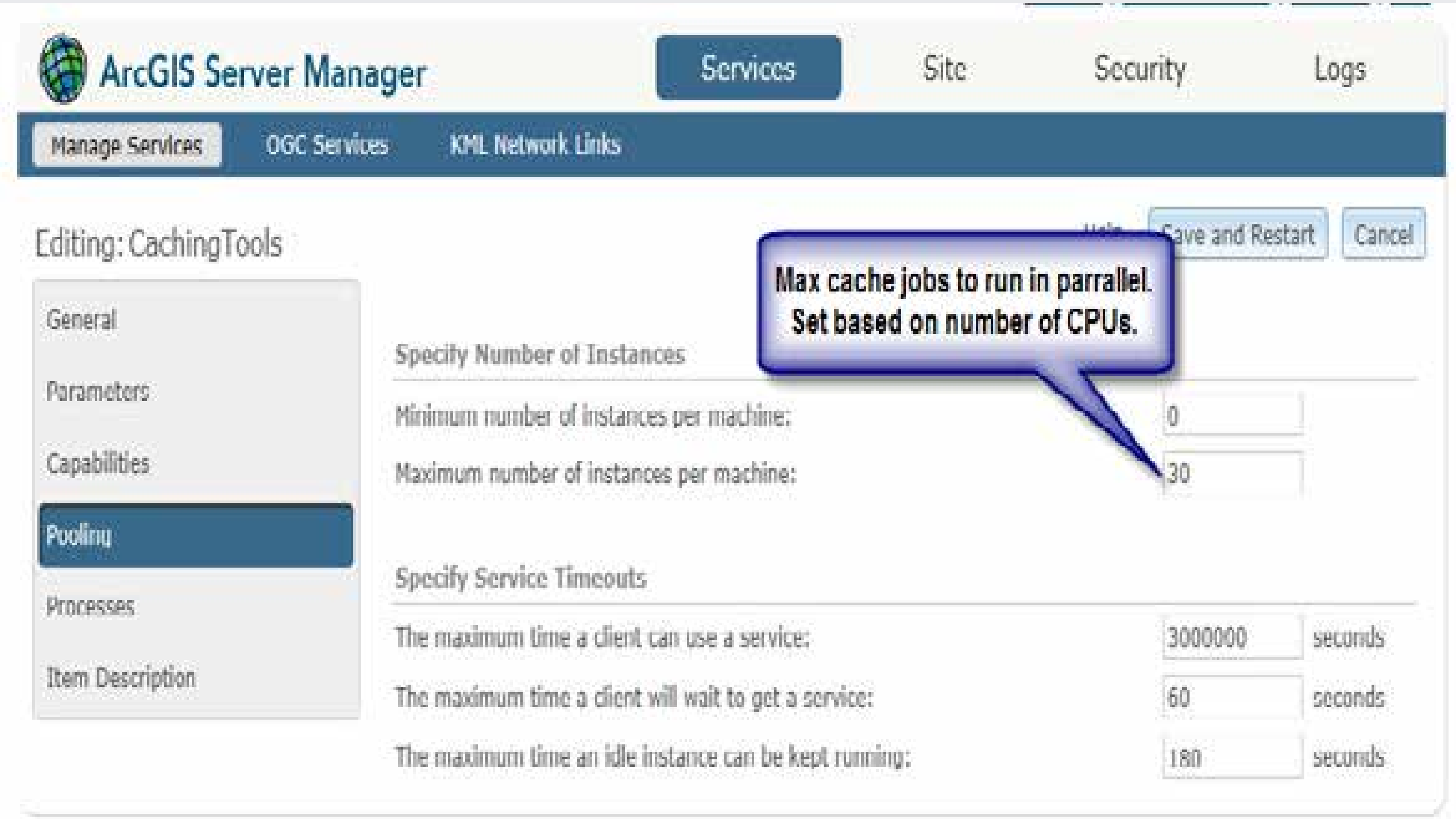
Use source data for resolutions below this value.
Build cache from lower cache levels for all lower resolutions.

Vistronix, Inc®

Building Cache from Image Services

- Publish Mosaic Datasets as Image Services.
- In ArcGIS10.1 you can build Cache Tiles directly from a Mosaic Dataset published as an Image Service (no MXD needed).
- Cache for Image Services are built from bottom up (high resolution first).
  - ArcGIS uses the *Max Cell Size* parameter to determine whether to build cache from source data or from cache tiles at higher resolution. Set the Max Cell Size to slightly higher than source resolution to build all lower res. caches from higher res. cache.
  - It is faster to build cache from higher resolution cache than from source data. Source data files can be large – requiring more IO.
  - You do not need to build pyramids on source data if you are *not* going to serve out the source data as Image Service.

Imagery – Building Cache



Editing: CachingTools

Max cache jobs to run in parrallel.
Set based on number of CPUs.

Specify Number of Instances

Minimum number of instances per machine: 0

Maximum number of instances per machine: 30

Specify Service Timeouts

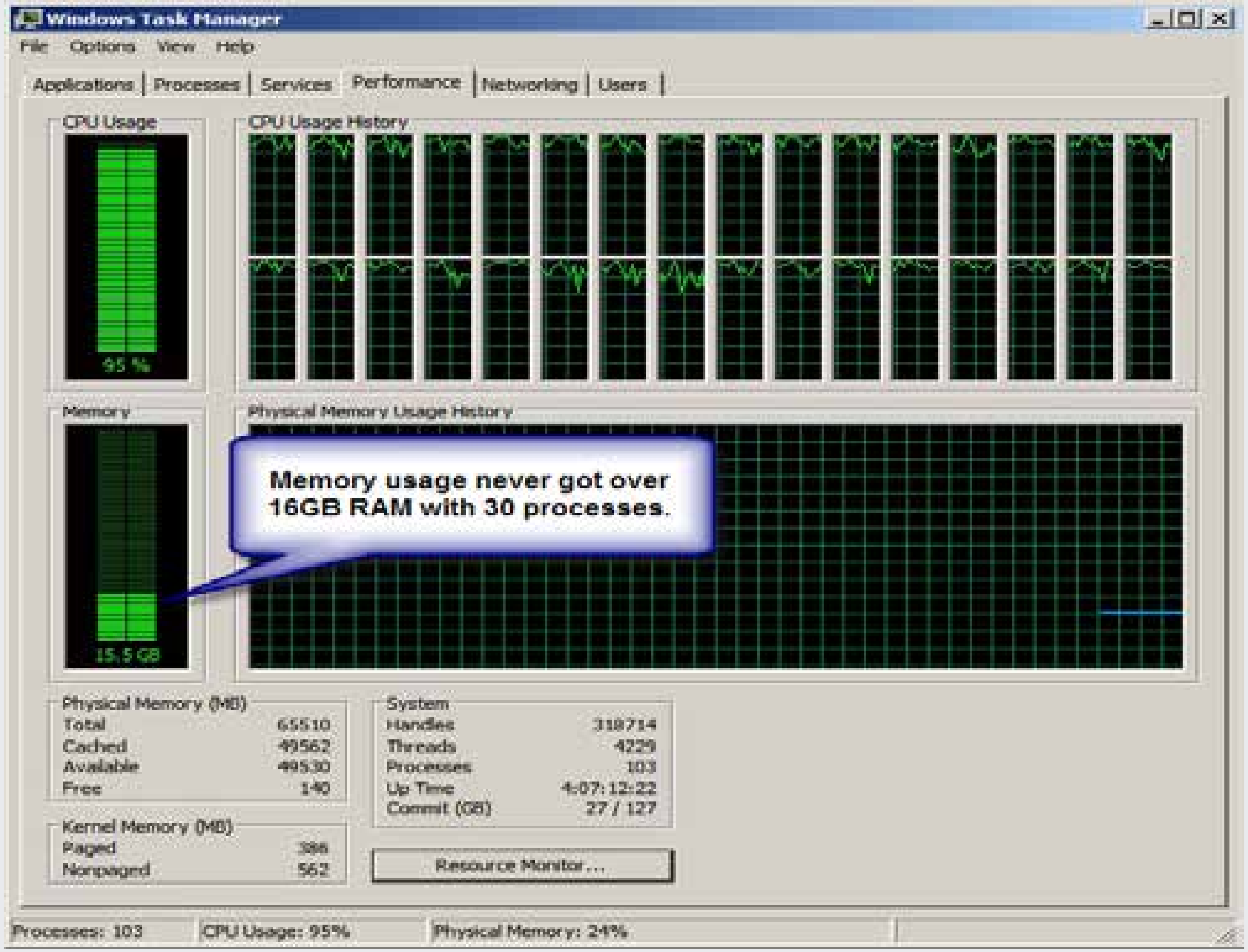
The maximum time a client can use a service: 3000000 seconds

The maximum time a client will wait to get a service: 60 seconds

The maximum time an idle instance can be kept running: 180 seconds

Vistronix, Inc®

Imagery – Building Cache



Physical Memory (MB)	
Total	65510
Cached	49562
Available	49530
Free	140

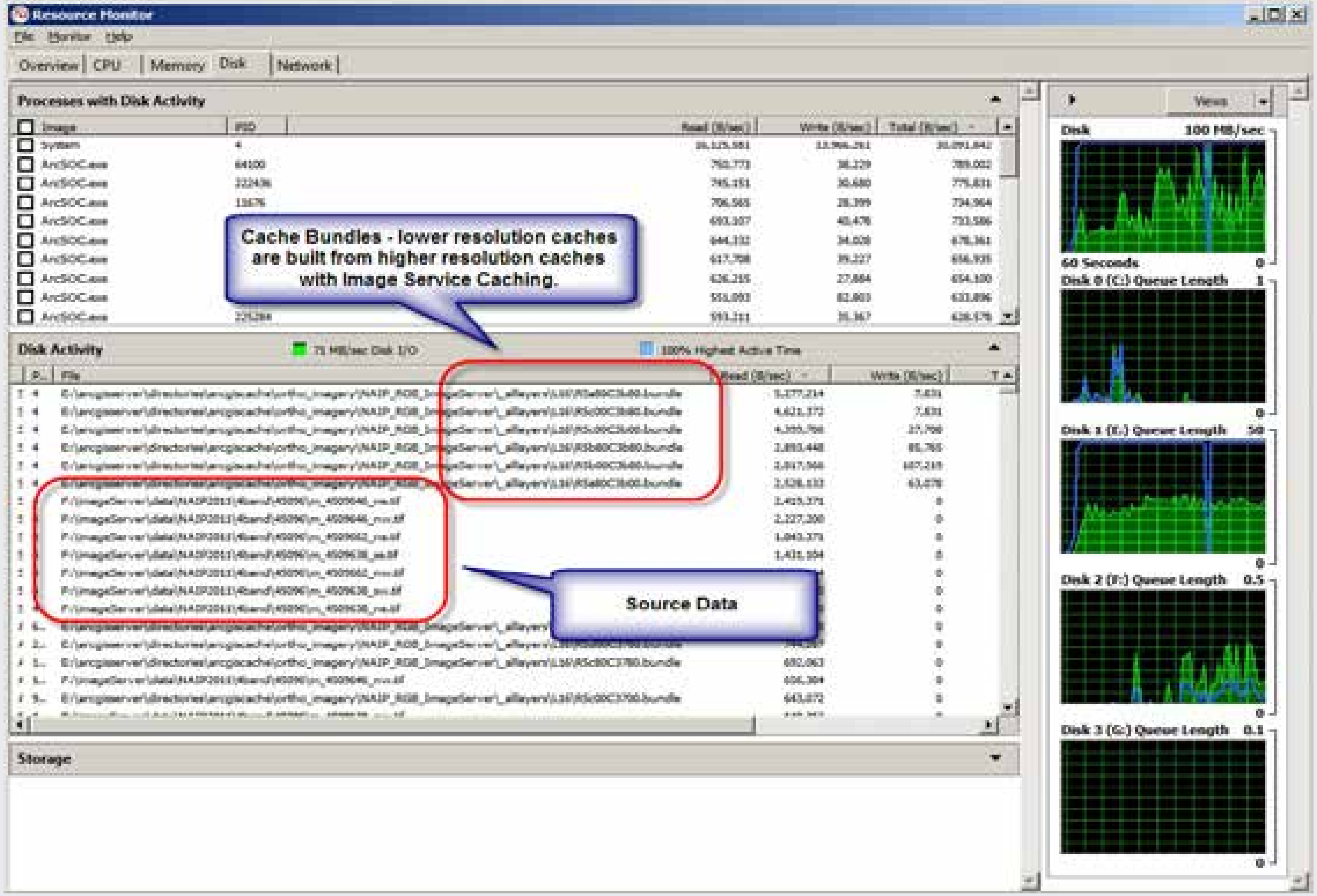
Kernel Memory (MB)	
Paged	386
Nonpaged	562

System	
Handles	318714
Threads	4229
Processes	103
Up Time	4:07:12:22
Commit (GB)	27 / 127

Processes: 103 | CPU Usage: 95% | Physical Memory: 24%

Memory usage never got over 16GB RAM with 30 processes.

Imagery – Building Cache



Resource Monitor
File Monitor Help
Overview CPU Memory Disk Network
Views
Disk 100 MB/sec
60 Seconds 0
Disk 0 (C:) Queue Length 1
Disk 1 (E:) Queue Length 50
Disk 2 (F:) Queue Length 0.5
Disk 3 (G:) Queue Length 0.1
Storage

Processes with Disk Activity

Image	PID	Read (B/sec)	Write (B/sec)	Total (B/sec)
☐ System	4	16,125,551	13,966,291	30,091,842
☐ ArcSOC.exe	64100	750,773	38,229	789,002
☐ ArcSOC.exe	222436	745,151	30,680	775,831
☐ ArcSOC.exe	11676	706,565	28,399	734,964
☐ ArcSOC.exe		693,107	40,478	733,586
☐ ArcSOC.exe		644,332	34,028	678,361
☐ ArcSOC.exe		617,708	39,227	656,935
☐ ArcSOC.exe		626,215	27,884	654,100
☐ ArcSOC.exe		551,093	82,803	633,896
☐ ArcSOC.exe	225284	593,211	35,367	628,578

Disk Activity 🟩 75 MB/sec Disk I/O 🟦 100% Highest Active Time

P...	File	Read (B/sec)	Write (B/sec)
4	E:\arcgisserver\directories\arcgiscache\ortho_imagery\NAIP_RGB_ImageServer_alllayers\L16\R5a80C3b80.bundle	5,277,214	7,831
4	E:\arcgisserver\directories\arcgiscache\ortho_imagery\NAIP_RGB_ImageServer_alllayers\L16\R5c00C3b80.bundle	4,621,372	7,831
4	E:\arcgisserver\directories\arcgiscache\ortho_imagery\NAIP_RGB_ImageServer_alllayers\L16\R5c00C3b00.bundle	4,355,756	27,758
4	E:\arcgisserver\directories\arcgiscache\ortho_imagery\NAIP_RGB_ImageServer_alllayers\L16\R5b80C3b80.bundle	2,893,448	85,765
4	E:\arcgisserver\directories\arcgiscache\ortho_imagery\NAIP_RGB_ImageServer_alllayers\L16\R5b00C3b00.bundle	2,817,566	107,215
4	E:\arcgisserver\directories\arcgiscache\ortho_imagery\NAIP_RGB_ImageServer_alllayers\L16\R5a80C3b00.bundle	2,528,133	63,078
	F:\imageServer\data\NAIP2011\4band\45096\m_4509646_ne.tif	2,415,371	0
	F:\imageServer\data\NAIP2011\4band\45096\m_4509646_nw.tif	2,227,200	0
	F:\imageServer\data\NAIP2011\4band\45096\m_4509662_ne.tif	1,843,371	0
	F:\imageServer\data\NAIP2011\4band\45096\m_4509638_se.tif	1,431,104	0
	F:\imageServer\data\NAIP2011\4band\45096\m_4509662_nw.tif		0
	F:\imageServer\data\NAIP2011\4band\45096\m_4509638_sw.tif		0
	F:\imageServer\data\NAIP2011\4band\45096\m_4509638_ne.tif		0
6...	E:\arcgisserver\directories\arcgiscache\ortho_imagery\NAIP_RGB_ImageServer_alllayers		0
2...	E:\arcgisserver\directories\arcgiscache\ortho_imagery\NAIP_RGB_ImageServer_alllayers		0
1...	E:\arcgisserver\directories\arcgiscache\ortho_imagery\NAIP_RGB_ImageServer_alllayers\L16\R5c80C3780.bundle	650,063	0
6...	F:\imageServer\data\NAIP2011\4band\45096\m_4509646_nw.tif	606,304	0
9...	E:\arcgisserver\directories\arcgiscache\ortho_imagery\NAIP_RGB_ImageServer_alllayers\L16\R5c00C3780.bundle	643,072	0

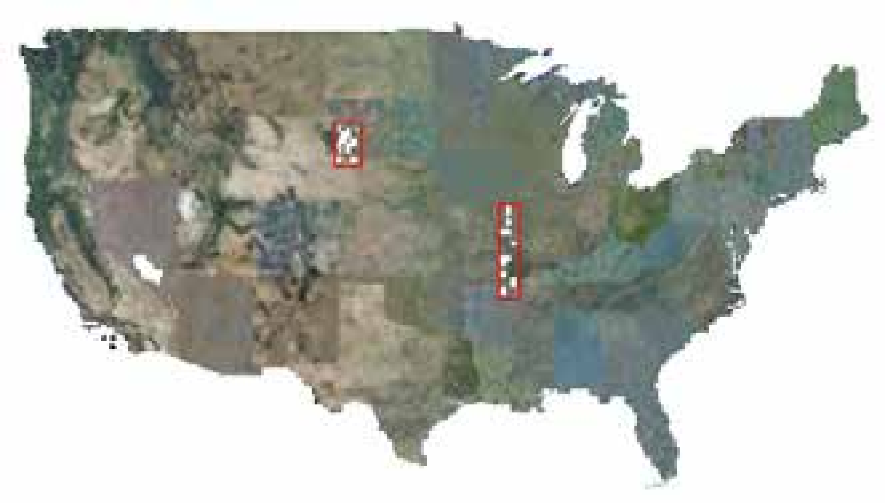
Cache Bundles - lower resolution caches are built from higher resolution caches with Image Service Caching.

Source Data

Vistronix, Inc®

Imagery – Building Cache

- 6 days later



- Holes?
- Because lower level caches are built from higher level cache, it is easier to spot holes. Missing tiles percolate up to the top of the pyramid for Image Services.
- What caused the holes?
  - –108 bad TIF files – not caught by Mosaic Dataset because headers were good, but files were somehow truncated.

Imagery – Building Cache

- Fix holes
  - Ran tiffinfo/python scripts to check all images.
  - Replaced bad imagery – 108 quarter-quads.
  - Re-ran cache for areas with holes.
  - Still had problems in some areas.
  - Used status.gdb to determine bundles with problems. Deleted “suspect” bundles, and re-ran.
  - 1 week to fix all problems.
  - Lesson learned -
    - Check source imagery before building cache.
    - It would have been faster to delete the cache layer and rebuild from scratch.

Imagery – Deploying Cache

- §Copy tiles to server.
- §Rename directories.
- §Create and publish “dummy” MXD.

At NRCS we chose to only publish the tiles – not the raw imagery - in order to save space. This requires that you create a dummy MXD with a simple vector with dimensions that match the cache tile envelope (CONUS outline).

The cache tile directory needs to be renamed to match the folder naming structure of the MXD based service.

Example: ortho_imagery_naip\layers

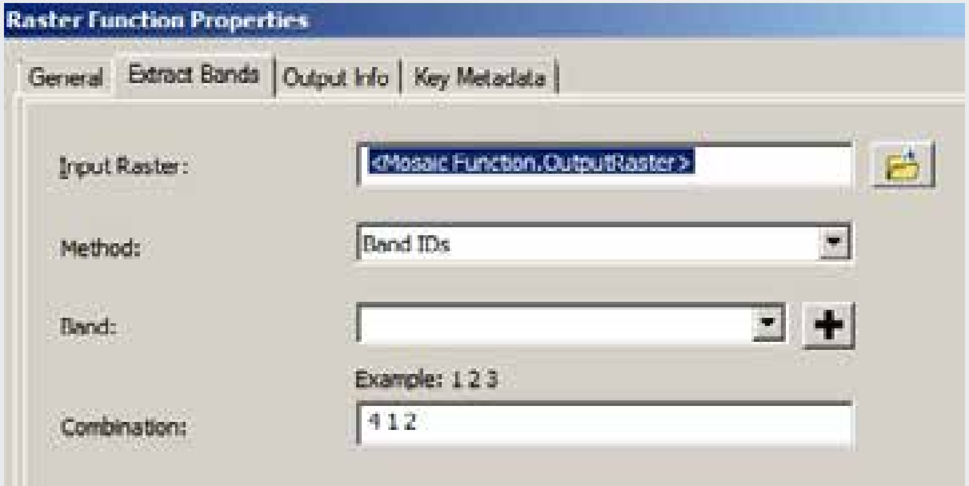
```
    folder-name_service-name\layers
```

NAIP – Tile Directory Sizes

Level	Resolution	Scale	Size on Disk
10	1:577,790	152 m	.197 GB
11	1:288,895	76 m	.654 GB
12	1:144,447	38 m	3 GB
13	1:72,223	19 m	9 GB
14	1:36,111	9.5 m	34 GB
15	1:18,055	4.7 m	132 GB
16	1:9,027	2.38 m	473 GB
17	1:4,513	1.19 m	1,510 GB

NAIP Imagery – Color Infra-Red

- Create a Reference Mosaic Dataset
- Add a function to specify band 4, 1, 2



Raster Function Properties

General | Extract Bands | Output Info | Key Metadata

Input Raster: <Mosaic Function.OutputRaster>

Method: Band IDs

Band:

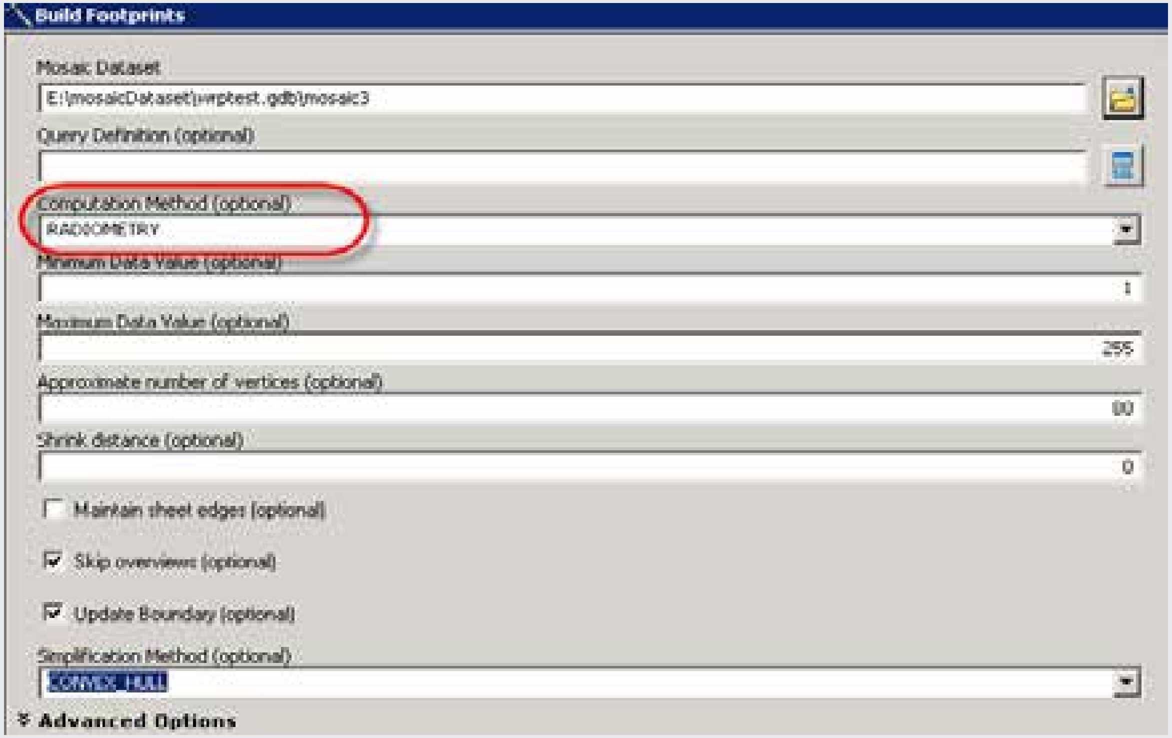
Example: 1 2 3

Combination: 4 1 2

- Publish the Reference Mosaic Dataset
- Build Cache

Imagery – Stewardship Land Imagery – SLI

- Sparsely populated high resolution imagery.
- Remove the NODATA value in the header and use Radiometry tool to fix footprints and adjust boundaries.



Build Footprints

Mosaic Dataset
E:\mosaicDataset\vwrptest.gdb\mosaic3

Query Definition (optional)

Computation Method (optional)
RADIOMETRY

Minimum Data Value (optional)
1

Maximum Data Value (optional)
255

Approximate number of vertices (optional)
80

Shrink distance (optional)
0

□ Maintain sheet edges (optional)

☑ Skip overviews (optional)

☑ Update Boundary (optional)

Simplification Method (optional)
CONVEX_HULL

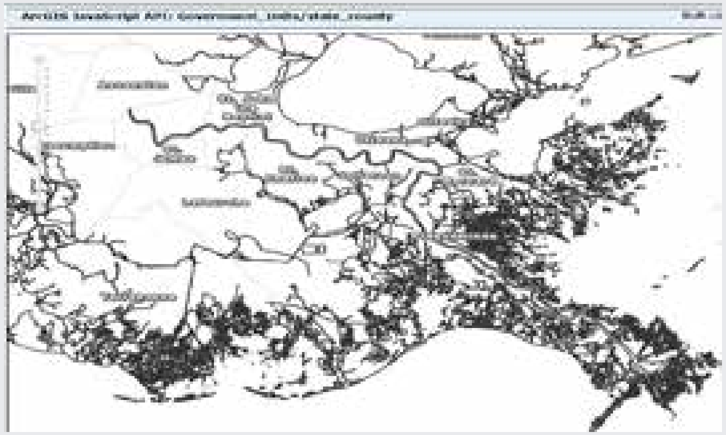
Advanced Options

Cache for Vector Layers

- Cache complex vector layers. The more complex, the greater the need for cache tiles. Labeling also has a severe impact on Web Map Services for vector layers.
  - –National Hydrography Dataset – very large and very detailed vector layer.
  - –Roads – multiple layers with complex labeling.
  - –State / County – high usage with complex polygons around the coast and water ways.

Imagery – Deploying Cache

- Why cache State / County boundaries? It is only 30MB.
- High activity (needed by all applications and users).
- More complex than you would think. Generalized layers can be used at low resolution, but detailed features are needed at high resolution. Louisiana has over 500,000 vertices. Server would need to query and render Map Image every time a user panned around Louisiana coast.



Cache for Vector Layers

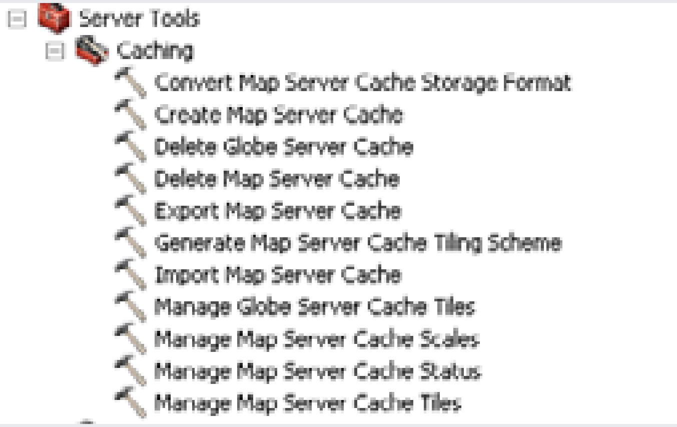
- NGCE decided to create transparent vector layers that could be draped over other layers (i.e. imagery).
- Used PNG32. PNG (which is a mix of PNG8 and PNG32) did not look very clean. Note: cache never looks exactly like non-cached services. Test and check.
- Symbology is difficult to design since the layers can be draped over imagery with very different coloration.
- Anti-aliasing and masking helps greatly with labeling.
- Source data was loaded to File Geodatabases.
- Generalized layers were used for lower resolution scales to improve cartography and speed up the caching process.

Cache for Vector Layers

- Symbology and labeling are challenging with vector layers. Issues: color, size, placement, halos. There are also issues with duplicate labels (on tile boundaries).
- Run sample areas at all resolutions.
- View tiles on top of sample imagery – make sure to check against different geographies, i.e. dark green vegetation, desert areas, urban areas and water ways.
- Base layers can be created by building mashups of vector layers over background imagery.

Tools

- ArcGIS Server 10.1 has quite a few tools for managing cache tiles



Server Tools
- Caching
  - Convert Map Server Cache Storage Format
  - Create Map Server Cache
  - Delete Globe Server Cache
  - Delete Map Server Cache
  - Export Map Server Cache
  - Generate Map Server Cache Tiling Scheme
  - Import Map Server Cache
  - Manage Globe Server Cache Tiles
  - Manage Map Server Cache Scales
  - Manage Map Server Cache Status
  - Manage Map Server Cache Tiles

- Caches can be directly viewed in ArcMap and ArcCatalog on the server by opening the cache folder with ArcMap/ArcCatalog.
- There is a geo-processing tool for creating Tile Packages (TPK) for offline access to cache tiles for specific areas and scales.
- The import tool can be used to replace tiles for areas and scales of a cache. Note: bug in export tool in 10.1 – fixed in 10.2.

Summary

- Space requirements are directly proportional to the extent of the dataset and the number of scales.   Multiply current size by 4 to determine total size for building an extra scale.
- Only build cache for the area you need and scales to support application and input dataset. (It does not make sense to create 1 meter resolution cache from 2 meter source data).
- Time requirements are directly proportional to the extent of the dataset and the number of scales divided by the number of CPUs. If you do not have a large SMP workstations then use multiple machines in a cluster.
- Use low-end DAS or large internal drives to save on cost of storage.
- Design, test and view at different scales over different backgrounds before committing 2-3 weeks of run time.
- Make sure to clear local cache when checking a cache layer so you can be sure you are looking at the new version of your tiles.
- Allocate time in your schedule to fix problems and to rebuild parts of cache.
- Cache Tiles are well worth the time and disk space!!!!

Questions