Agenda

• Advanced topics
  - Best practices
  - Update map caches
  - Caching architecture
  - Performance considerations
  - Caching in the Amazon cloud
  - Recovering from cache failures
Best Practices
Understanding cache structure

Supertile: 4096 pixels, 16 x 16 tiles

Standard tile: 256 x 256 pixels

Total tiles: 16,384
Supertiles and Labeling

- **ArcGIS Server Draws Large Areas**
  - Helps Reduce Duplicate Labels
  - Large Areas are Supertiles
  - Divided into Individual Tiles

- **Duplication May Occur**
  - Label Engine Not Aware of Labels on Adjacent Supertiles
  - Use Annotation or MapPlex Labels with Rules
  - Use Map Server Cache Tiling Scheme To Polygons
How much of a cache should you generate?

- **Cache by feature**
  - Geographic elements
  - Generates tiles for intersecting supertiles

- **Saves on...**
  - Generation time
  - Processor resource
  - Disk usage

**NM highway case study:**
Build 20 of 64 supertiles for the bundle shown
Creating a test cache

- Select test area location with most geographic complexity
- Create a simple feature class covering the test area
- Generate tiles at all scale levels based on this feature
Setting the Number of Instances

- Cache Tools Geoprocessing Service
  - Start with N + 1
    - N = the total number of CPU’s per server in your site (workers)
    - 1 controller
  - Example: x2 - 4 Core GIS Servers/Site = 8CPU + 2 = 10 Instances
  - Max Instances should always be >2

8CPU’s+1= 9 Instances

8 CPU
Controlling the # of instances for a job

- CachingTools service controls caching instance per machine
- Manage Map server Cache Tiles controls # of instances per job
  - Set to -1 to use all instances
Strategic cache creation

- Feature class covers about 25% of California
- Contains about 97% of California’s population
- Caching with this layer saves:
  - 943,000 tiles
  - 9.3 GB of space
  - 17 hours of caching time
- Bing analysis of access
  - [http://hotmap.msresearch.us](http://hotmap.msresearch.us)
The ideal tiles to cache on demand

- Few simple features
  - Barren homogenous area
  - Rarely accessed
- Draw relatively fast
- Large scales only
ArcGIS Server Descriptive Tile Analysis

- **Tile Usage Heat map**
- **ArcGIS Server descriptive tile analysis**
More tips for caching by feature class

- Map is automatically published in WGS 84 Web Mercator with Auxiliary Sphere
  - You can create your own tiling scheme

- Avoid numerous small features
  - Aggregate Polygons tool
  - Dissolve tool

- Avoid excessive vertices
  - Simplify Polygons tool

**Tips for caching by feature class**

blog post
Pre-create coupled with cache on demand

- Pre-create high use areas
  - Population centers
  - Parks, roads, attractions

- Features
  - Cover popular extent
  - Dissolve and generalize
  - Generate key tiles
  - All others generated on demand
Handling tiles you do NOT create

• Create “No Data” tile
  - Same image format (JPG or PNG)
  - Same size (256 x 256)
  - Save in cache folder
    …\dataframe\_alllayers

• Knowledge base article 36939 has sample files
Updating Map Caches
Updating a cache

- Architecture
  - Using a staging server
  - Using a cluster

- Implementation
  - Geoprocessing
Cache update strategies

- Rebuild the entire cache
  - Size of cache
  - Time to cache
- Rebuild specific tiles
  - Rebuild at specific scales
  - Rebuild areas based on change detection
Cache Update Automation

- Use Model Builder to script Cache Update Automation
  - Rebuild Specific Tiles
  - Export to Python
  - Schedule Run Time

Compare feature classes
Show edits since reconcile
Strategic cache updates
System architecture for caching
Update a cache using a staging server

Staging ArcGIS Server Instance

Map service

All layers for cartography of map service

Cache folder

Production ArcGIS Server Instance

Map service

Layers for TOC and Query

Cache folder

Copy cache

(On-demand caching needs the full map to build the cache)

State of Indiana Imagery With Metadata
Update a cache and data using a staging server

Staging ArcGIS Server Instance

- All layers for cartography of map service

Map service

Cache folder

Geodatabase Replication

Multi-user versioned Geodatabase

Production ArcGIS Server Instance

- Layers for TOC and Query

Map service

Cache folder

Geodatabase (File or Multi-user)
Multi-Machine Deployment With Web Adaptor & Clusters

- Organize GIS Servers into Groups or Clusters
  - Run subset of services on clusters
  - Generate Cache

Generate Cache or serve GP tasks on cluster 1
Serve map services & cache on cluster 2
Generate using multiple LAN hosts

- Configure instances for CPU’s on all SOC hosts
- Enable “Use local cache directory…”
Caching in the Amazon Cloud
ArcGIS Server Cloud Builder
for Amazon Web Services

- Quickly provision Amazon Web Service Instances
  - Linux - This Is Awesome !!!!
  - Windows
- Supports Auto Scaling
  - Set capacity thresholds
    - Scale up new EC2 instances when needed
    - Scale down EC2 instances when finished
- Did I mention Linux?
Building cache in the cloud

- Likely it is more efficient to build the cache in the cloud
  - Transfer options: ArcGIS Server, RDP, S3, FTP, AWS Import/Export
- Use Large instances
  - High-memory instances
- Auto-scaling?
  - Maybe if you can’t monitor the job very closely
Building cache in the cloud
Copying caches

- **Compact cache**
  - RoboCopy or RichCopy

- **Exploded cache**
  - Windows Copy/Paste inefficient for large caches
  - Use RichCopy or RoboCopy command
    - Use Multiple Directory Copy Threads
    - `/D` switch will only copy new tiles

*Multiple Threads: 40  **Directory Copy Threads: 40*

<table>
<thead>
<tr>
<th>Cache Type</th>
<th>Size on Disk</th>
<th>Size</th>
<th>Number of Files</th>
<th>RoboCopy*</th>
<th>XCopy</th>
<th>Windows Copy/Paste</th>
<th>Rich Copy**</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compact Cache</td>
<td>1.43 GB</td>
<td>1.43 GB</td>
<td>115</td>
<td>2:08</td>
<td>2:28</td>
<td>2:26</td>
<td>2:12</td>
</tr>
</tbody>
</table>
Performance Considerations
Disk maintenance

- Turn off backup
- Turn off virus scanning
HTTP connections

• Turn On HTTP KeepAlive for best performance
  - Multiple image requests don’t need to open a new connection
  - Biggest benefit with one host (~2-3 times faster tile requests)
• Apache: KeepAlive directive
• IIS
Cache distribution

- HTTP 1.1 spec constricts browsers to two simultaneous downloads

- Improve cache retrieval performance by using multiple hosts
  - Can be one server with multiple DNS entries (cache1.mydomain.com, cache2.mydomain.com)

- Geographically distribute ArcGIS Server instances
Using multiple domains

- With multiple services
  - Use a different domain for each service

- With one service
  - API’s support multiple web services endpoints for a single layer

```javascript
var layer = new esri.layers.ArcGISTiledMapServiceLayer(
  "http://www.mydomain.com/ArcGIS/rest/services/myservice/MapServer",
  { tileServers: [
    "http://cache1.mydomain.com/ArcGIS/rest/services/myservice/MapServer",
    "http://cache2.mydomain.com/ArcGIS/rest/services/myservice/MapServer"
  ] });
```

- Use with small cache tiles
- Can reduce browser caching and result in more HTTP connections
Switching to a Dynamic Request

- Supported by Map Service & the Web API’s
- Force a Cached Service to Render Dynamic
  - Use ArcGISDynamicMapServiceLayer(Cached Service URL)
- Render cached tiles from smaller scales levels 0 – 15
- Render dynamic services at larger scale levels 16 – 19
  - Large scales = less area to draw features = faster

Scale Levels 10 – 17 Cached

Scale Levels 18 – 19 Forced Dynamic Draw

Java Script API – Dynamic Draw
Recovering from Cache Failures
Cache failure & recovery – Out of the box

- Out of the box tools
  - Course Grained for locating extents
  - Fix errors re-caches extents where errors were reported
Cache failure & recovery – sample tools

- Cache Validation Tools
  - Fine grained for locating tiles by file size
Other Caching Resources

- Compare feature classes
- Show edits since reconcile
- Derive map service statistics from the ArcGIS Server logs
- Write requested map extents to a feature class
- Tile Usage Heat map
- ArcGIS Server descriptive tile analysis
- Web GIS Blog
Relevant Sessions

• **Publishing Image Services in ArcGIS**
  - Thu 7/26/2012 -- 8:30 AM - 9:45 AM Room 28D

• **Advanced Map Caching Topics**
  - Thu 7/26/2012 -- 8:30 AM - 9:45 AM Room 10

• **Best practices for caching imagery in services**
  - Thu 7/26/2012 -- 11:00 AM - 11:30AM Demo Theater

• **Publishing Image Services in ArcGIS**
  - Thu 7/26/2012 -- 3:15 PM - 3:35 AM Room 28D

• **Map Caching: Tips from ArcGIS Online Team**
  - Wed 7/26/2012 – 3:15 PM – 3:35 PM Room 01B
Steps to evaluate UC sessions

- My UC Homepage > “Evaluate Sessions”
- Choose session from planner
  OR
- Search for session

www.esri.com/ucsurveysessions
• Thank you for attending
• Have fun!
• Open for Questions

• Please fill out the evaluation:

  www.esri.com/ucsessionssurveys

  First Offering ID: 667
  Second Offering ID: 1922