Geodatabase Tuning and Performance

Gillian Silvertand
Greg Cunningham
Workshop outline

- Performance and scalability
  - Description
- Design
- Configuration
- Tuning and optimization
- Maintenance
Performance

- Response time for end user
- Total response time:
  - Based on load, number of users, number of operations

![Diagram showing the flow of data and processing in a multiuser geodatabase system.](Image)
Scalability

◆ The capacity to handle increasing load or demand
◆ Ability to:
  ◆ Maintain response times
    • Within desirable, defined parameters
  ◆ Change size / configuration for conditions
  ◆ Continue to function well
    • When system is changed by size or volume
Components of scalability

◆ Scalability is comprised of:
  ◆ Response times
    • End user perception of redraws
    • Editing
    • Queries
  ◆ Throughput
    • Amount of data / unit of time
  ◆ Load
    • Predefined number of client connections
    • Predefined connection mix (editors, viewers, etc.)
Bottlenecks

- **Server Hardware**
  - Ensure storage, CPUs, memory requirements are met
    - ESRI Systems Integration Group
  - Server O/S must be adequately configured

- **Network**
  - Easily becomes the bottleneck
  - Ensure network requirements are met

- **DBMS**
  - Must be adequately tuned specifically for GDB
    - See – Configuration and Tuning Guides

- **ArcSDE**
  - Minor tuning required

- **Geodatabase**
  - Is data model used scalable?

Very dependent on DBMS performance
Bottlenecks cont.

- **ArcGIS client**
  - Somewhat configurable
    - Example – selection threshold
- **ArcObjects Customization**
  - Design and delivery of functionality must be carefully considered
  - ArcObjects delivers extremely rich set of functionality
  - Too much functionality delivered to client:
    - Could overwhelm client machine
    - When numerous clients interact with server, server could be overwhelmed
  - Customization **scalability** must be in the **software design** document
Maximize performance and scalability

◆ Internal:
  ◆ Reducing database work
    ◆ Reusing shared cursors (shared / held cursors)
    ◆ Fewer redo entries
    ◆ Fewer commits
  ◆ Better client side caching
    ◆ Fewer round-trips
◆ External:
  ◆ Better Geodatabase / ArcSDE integration
    ◆ Better workflow design
    ◆ Better Geodatabase design
  ◆ Optimized ArcObjects code
  ◆ Optimized map use
Internal performance enhancements

- Better integration with ArcSDE
- Binding frequently executed SQL statements
- Pre-allocating row id's
- Cleaner transaction model
- Improved schema caching
- Open document optimization for complex map documents
- Reduction of round-trips with describe cache
- Cursors used extensively
Workshop outline

- Performance and scalability
- Design
  - Geodatabase design issues
- Configuration
- Tuning and optimization
- Maintenance
Geodatabase integrity

- Business rules modeled as geodatabase behavior
- Rules stored in the database, enforced by ArcGIS
  - Domains enforce database integrity
  - Relationships enforce referential integrity
  - Subtypes enforce user-defined integrity

<table>
<thead>
<tr>
<th>DBMS</th>
<th>Geodatabase</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECK constraint</td>
<td>Domain / Subtype</td>
</tr>
<tr>
<td>Primary key</td>
<td>Relationship class</td>
</tr>
<tr>
<td>Foreign key</td>
<td></td>
</tr>
<tr>
<td>User-defined procedure</td>
<td>Topology, geometric</td>
</tr>
<tr>
<td></td>
<td>network, network dataset</td>
</tr>
<tr>
<td>Trigger</td>
<td>Composite relationship class</td>
</tr>
</tbody>
</table>
Feature datasets

◆ Feature dataset
  ◆ Groups feature classes together
  ◆ Required for topology and geometric networks

◆ Performance considerations:
  ◆ All feature classes in a feature dataset are instantiated
    • Results in more activity on server than actually may be necessary
  ◆ Map cache built for all feature classes in a feature dataset

◆ Design considerations:
  ◆ Privileges granted to entire contents of feature dataset
    • Cannot grant to individual feature classes within feature dataset
Subtypes and domains

- **Subtypes:**
  - Yield better performance than subdividing data
    - More feature classes results in more queries – one query for each feature class displayed
    - DBMS more capable of handling very large tables vs. high volume of concurrent SQL
  - Caveat – automatically symbolized by subtype

- **Domains:**
  - Range – negligible performance impact
  - Coded value – minor performance impact during edit session
  - Domain cached on client
Relationship classes

- An association between objects
  - Feature to row, feature to feature

- Performance considerations:
  - Composite relationships have higher cost
  - Relationship classes result in additional roundtrips to the server
  - Relationship class always turned on
    - More efficient to use temporary join or relate
  - Use for maintaining integrity in an edit session
Annotation and labeling

- **Annotation:**
  - Stored in map document or geodatabase
  - Can be feature-linked
- **Labeling:**
  - Not stored in the geodatabase
- **Performance considerations:**
  - Labels and annotation query the server
  - Labels can degrade performance – need to find best placement
  - **Consolidate** annotation to improve performance
    - Reduces number of queries
Spatial relationships

- **Geometric networks:**
  - Performance will degrade with many feature classes
    - Degradation dependent on hardware and network resources
  - Geodatabase has to rebuild connectivity when editing
  - All feature classes in the network will be returned

- **Network datasets and Topology:**
  - Behavior on request
    - Built (Network Dataset) or Validate (Topology) as needed
  - Very little associated overhead
    - Like simple feature classes
Compression statistics

- Applied per raster tile
  - Determines row sizes in raster block (BLK) table

- Expressed as percentage of input
  - Cannot predict behavior without loading sample

\[
\text{Compression } \% = \left( \frac{\text{input}_\text{size} - \text{output}_\text{size}}{\text{input}_\text{size}} \right) \times 100
\]

Source data

- Pixel depth: 8-bit
- Number of bands: 1

<table>
<thead>
<tr>
<th>Compression</th>
<th>Avg Row Length</th>
<th>Size (MB)</th>
<th>Compression %</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
<td>16,488</td>
<td>262</td>
<td>-174.63</td>
</tr>
<tr>
<td>LZ77</td>
<td>13,568</td>
<td>130</td>
<td>-36.27</td>
</tr>
<tr>
<td>JPEG (75)</td>
<td>3,868</td>
<td>36</td>
<td>62.26</td>
</tr>
<tr>
<td>JPEG (50)</td>
<td>2,616</td>
<td>24</td>
<td>74.84</td>
</tr>
<tr>
<td>JPEG (25)</td>
<td>1,749</td>
<td>16</td>
<td>83.23</td>
</tr>
<tr>
<td>JP2 (200)</td>
<td>8,030</td>
<td>91</td>
<td>4.61</td>
</tr>
<tr>
<td>JP2 (130)</td>
<td>1,984</td>
<td>18</td>
<td>81.13</td>
</tr>
<tr>
<td>JP2 (60)</td>
<td>284</td>
<td>3</td>
<td>96.86</td>
</tr>
</tbody>
</table>

No pyramids

Copyright © 2005 ESRI. All rights reserved.
Pyramid overhead

- Pixel size doubles at each pyramid level
  - Results in ¼ as many pixels
  - Tiled and stored in BLK table
- Theoretical overhead is 33%
- ArcSDE computes optimal number of pyramids
  - Based on width and height of raster in pixels
Workshop outline

- Performance and scalability
- Design
- Configuration
  - ArcSDE connections
  - ArcSDE logfile tables
- Tuning and optimization
- Maintenance
**ArcSDE connections**

- **Application Server Connection**
  - Light-weight clients
  - Server memory and/or CPU contentions are not an issue
- **Direct Connect**
  - Reduce memory and CPU contentions on server
Example – ArcSDE network traffic

- Compare amount of data transferred over network for:
  - Application Server
  - Direct Connect
  - UNC file sharing

- Example – ArcSDE 9 and ArcIMS 9 HTML browser

Image 1 (vector only)
- Application Server: 137 KB
- Direct Connect: 179 KB
- UNC file sharing: 1124 KB

Image 2 (vector and image)
- Application Server: 542 KB
- Direct Connect: 686 KB
- UNC file sharing: 116095 KB
Performance enhancement results

- **SQL Count** greatly reduced in 9x
- Parse once, execute many times
- Increases **scalability** of geodatabase
- Increases **performance** of geodatabase

<table>
<thead>
<tr>
<th>Extents Queried</th>
<th>Scale</th>
<th>ArcGIS 8.3 Ave. display time (sec.)</th>
<th>ArcGIS 9 Ave. display time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>395</td>
<td>3.57</td>
<td>2.75</td>
</tr>
<tr>
<td>100</td>
<td>791</td>
<td>4.42</td>
<td>3.90</td>
</tr>
<tr>
<td>100</td>
<td>1582</td>
<td>5.57</td>
<td>4.99</td>
</tr>
<tr>
<td>100</td>
<td>2373</td>
<td>7.64</td>
<td>5.99</td>
</tr>
</tbody>
</table>
ArcSDE logfile tables

- Scratch tables
  - Maintain sets of selected records
- Used by ArcSDE client applications:
  - ArcGIS: Created when user's first selection is >= 100 features
  - Versioning
    - Reconcile and post
    - Disconnected editing checkout
- Alter SERVER_CONFIG table
- Architecture choice based on use
Logfile choice

- Must determine the appropriate architecture based on:
  - DBMS
  - DBMS users
    - Multiple DBMS users
    - Single DBMS users with multiple sessions
  - Types of selections created
    - Large
    - Small
  - Scalability requirements
    - Reuse SQL
    - Recycle pool (Oracle 10g)
Index fragmentation

◆ It is all about indexes

◆ Things to look for:
  ◆ Slower performance between compresses
  ◆ Massive edits
  ◆ Check clustering_factor

◆ Take “good” traces / snapshots to compare
  ◆ tkprof, Profile, Event Monitor, Onstat

◆ Trace “bad” and review execution plans for change

◆ ArcSDE does not coalesce or rebuild after compress
  ◆ Rebuild, coalesce or defragment indexes
  ◆ Maintain index statistics
Index management

- Updating, inserting and deleting causes fragmentation
  - Indexes can become unbalanced
  - Fragmentation leads to increased I/O
  - Fragmentation leads to performance issues
- Rebuild indexes
  - Reduces logical I/O
  - Rebuild – reorganizes index
    Sdetable -o rebuild_indexe
- Maintain index statistics
Workshop outline

- Performance and scalability
- Design
- Configuration
- Tuning and optimization
  - Application
- Maintenance
- Summary
Desktop optimizations

“The best way to improve performance of something is... to not do it!”
Cary Millsap (Hotsos)
Query order

- Spatial or attribute first query support

- Use:
  - Time series
  - Attribute selectivity better than spatial
  - Historical databases with stacked features
Group features for rapid display – sdegroup

- Fewer large features draw faster than many small ones
  - Less attributes and geometry metadata
  - Better coordinate compression
- Group spatially or by attribute

77,773 single-part features
12 multi-part features grouped spatially

100 multi-part features grouped by attribute
Scale dependent layers

- **Reduces** ArcSDE I/O
  - **Control visibility based on** map extent

- **Example – lightning layers**
  - **Display** WILSON\_LIGHTGROUP when zoomed out
    - Few features to fetch
    - General background data
  - **Reveal** WILSON\_LIGHTNING when zoomed in
    - Spatial filter allows fast fetch
    - Enables interaction with individual points
**Symbology**

- **Use simple symbols**
  - Reduces CPU usage on the client machine

- **Avoid:**
  - Hatched symbols for polygons
  - Complex line symbols
  - Complex point symbols

- **Use separate documents for:**
  - Editing and analysis
  - Printing and map publishing
Simple vs. complex symbology: Case Study

- **Simple map document:**
  - Display count | Total time/per display (s)
  - 10 displays   | 15

- **Complex map document:**
  - Display count | Total time/per display (s)
  - 10 displays   | 31
ArcMap map cache

- Caches features on the client
  - Used in ArcMap (editing and viewing)
  - Reduces number of spatial queries against server
    - Impacts scalability
    - Improves performance
    - Reduces server interaction
- Performance improvement, especially for:
  - Complex map documents
  - Multiple label classes
- User manages the map cache
  - Rebuild for Area Of Interest or use the Auto-Cache option
ArcMap document optimization

- Visible layers in TOC returned from server
  - Make newly added layers not visible by default
  - Uncheck layers when not needed
- Only include necessary feature classes
  - Include all related classes
- All layers selectable by default
  - Set Selectable Layers – avoids more queries
Annotation vs. labels

- **Labels placed on the fly**
  - Extra placement processing
  - Fetch attributes from server
  - No additional storage
- **Annotation stored as features**
  - Can consolidate into annotation classes
- **Performance**
  - Many label classes impact draw time and scalability
  - Feature linked annotation introduces additional performance impact
Using multiple label classes – example

- Dozens of label classes – one feature group layer
Joins processed by server

◆ Types
  ◆ ArcMap join with:
    ◆ Keep only matching records option, and…
  ◆ ArcSDE view
  ◆ DBMS view

◆ Benefits
  ◆ Best performance
    ◆ Use DBMS join algorithms
    ◆ Data manipulated in server memory
User defined indexes

- Indexes should be created on:
  - Columns used in join operations
  - Columns queried frequently
  - Columns with high % of uniqueness

- Composite indexes:
  - Multiple columns
  - WHERE clause references all or the leading portion of the columns
    - Most commonly accessed or most selective first

- Create indexes using:
  - ArcCatalog
  - ArcSDE commands
  - Oracle
Demonstration

Optimizing ArcMap documents
Workshop outline

- Performance and scalability
- Design
- Configuration
- Tuning and optimization
- Maintenance
  - Multiuser database
- Summary
Versioning performance

◆ Will my multiuser geodatabase scale indefinitely?
◆ Performance can degrade over time
    ◆ Workflow can often generate unnecessary versions
        • Versions may just “hang-out” on the state tree
        • Prevents state tree from being fully compressed
    ◆ Response time increases with volume of states
        • In production, number of rows can increase significantly
        • Table growth is a function of edits
    ◆ Performance lags introduced with stale statistics
        • Database might choose a sub-optimal execution plan
Essential to devise workflow

- Versioning supports a variety of workflows
  - Workflow will vary by organization
- Maintain separate stages of a project
- Comply with business procedures
- Quality Assurance (QA)
  - Ensure timely and accurate database changes
  - Security – version permissions
    - Preserve geodatabase integrity
    - Vulnerability of versions to inadvertent modifications
- Maintain geodatabase performance
Performance as a function of states

- **state_lineages** table is very active
  - Depends on workflow and number of editors
    - Response time increases with volume of states per version
  - Can easily grow beyond a million records
    - More edits per lineage → higher logical I/O
      - Expensive to traverse lineage
    - Edits increase index size
  - Rebuild indexes
  - Update statistics using DBMS
    - Generate false statistics (Oracle)

<table>
<thead>
<tr>
<th>lineage_name</th>
<th>lineage_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

Copyright © 2005 ESRI. All rights reserved.
Performance of delta tables

- Size of delta tables can be a significant factor
  - Delta tables analyzed upon creation
    - Update statistics relative to frequency of edits
  - Pay attention to the D-table
    - An anti-join is used to return rows to client
      - Reverse logic from a join; return only rows that do not match
    - Primary key index on the D table should be rebuilt
Statistics

- **ArcSDE does not maintain statistics; DBA responsible**
  - Describe data for database optimizer
  - Critical for maintaining performance

- **Updating statistics**
  - Depends on editing activity
  - Before and after database compress

- **Use**
  - Analyze Components dialog
  - DBMS statements
  - ArcSDE command line
Version management

- Version creation dependent on workflow
  - Fewer versions (workflow)
  - Delete versions when not needed
- Compress to reduce depth of state tree
  - Outstanding versions may prevent states from being compressed
  - Maximize results of compress
    - Reconcile with DEFAULT
      - Brings versions under DEFAULT – helps compress
      - Pay attention to order in which versions are reconciled
    - Option – Reconcile order

As DEFAULT progresses, designs (D) are created at different states
State tree example

- Thousands of versions
  - Versioning model will scale
- Example – editors have own versions
  - Majority reconciled / posted to DEFAULT daily
    - DEFAULT located lower in lineage
    - Outstanding versions exist
- Example – design versions
  - Reconciled to DEFAULT after project complete
    - Active for project duration
    - DEFAULT located up in lineage
Compressing the database

- Maintains performance
  - Moves common rows from delta tables into base tables
  - Reduces depth of state tree
    - Removes redundant rows, unreferenced (save points) and orphaned states

Before

After

Copyright © 2005 ESRI. All rights reserved.

Geodatabase Tuning and Performance 49
Retaining rows in the delta tables

- Performance still good with rows in delta tables
  - Edits unique to version remain
    - E.g., Parcels version not reconciled and posted
- Required for workflow:
  - History
  - Work order scenarios
  - Long transactions
- Compressing all rows only when unregistering

Before

<table>
<thead>
<tr>
<th>Parcels</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>92</td>
</tr>
<tr>
<td>93</td>
<td></td>
</tr>
<tr>
<td>97</td>
<td></td>
</tr>
</tbody>
</table>

After

<table>
<thead>
<tr>
<th>Parcels</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>40</td>
</tr>
<tr>
<td>97</td>
<td></td>
</tr>
</tbody>
</table>

Copyright © 2005 ESRI. All rights reserved.
How often should you compress?

- Depends on the amount of editing activity
- Not unreasonable to compress every night
  - Routine maintenance for highly edited databases
- Compress at least once a week
  - Medium to low volume of edits
- `compress_log` provides information on compress
  - `start_time` and `end_time`
  - Number of states compressed
Version performance – example workflow

Suggested weekly or daily workflow:

1. Create versions
2. Complete edits (as per business process)
3. Build statistics on delta tables, versioning tables
4. Compress (optional – removes redundant states)
5. Reconcile and post versions (option – batch samples)
6. Delete versions (optional – if permitted by workflow)
7. Compress the geodatabase
8. Build statistics, re-build indexes
9. Re-create versions
Workshop outline

- Performance and scalability
- Design
- Configuration
- Tuning and optimization
- Maintenance
- Summary
Summary

◆ Your actions affect your performance!
◆ Remember to:
  ◆ Design efficient geodatabase architecture
  ◆ Follow DBMS tuning and configuration recommendations
  ◆ Optimize map documents
  ◆ Maintain indexes and statistics
  ◆ Design efficient versioning workflow
  ◆ Compress
Additional Resources at the Conference

◆ **Technical Workshops**
  - **Administration for Oracle**
    - Tuesday 1:30, Room 10 / Thursday 8:30, Room 10
  - **Administration for SQL Server**
    - Wednesday 8:30, Room 10 / Thursday 1:30, Room 10
  - **Geodatabase Tuning and Performance**
    - Wednesday 10:30, Room 1A / Thursday 3:30, Room 1A
  - **Integrating ArcSDE with Enterprise Databases**
    - Tuesday 3:30, Room 1A / Thursday 10:30, Room 1A

◆ **ArcSDE product island**

◆ **Doctor’s Office**
Additional Resources After the Conference

- **Documentation**
  - *Versioning* technical paper
  - *Versioning Workflows* technical paper
  - *Modeling and Using History in ArcGIS* technical paper

- **Instructor-led training**
  - Introduction to the Multi-User Geodatabase (2-day fall 2005)
  - Managing a Versioned Database (3-day)
  - ArcSDE Administration for Oracle (5-day)
  - ArcSDE Administration for SQL Server (5-day)
  - ArcGIS Enterprise Systems – Performance and Scalability

- **Web sites**
  - ESRI Technical Support – http://support.esri.com
GIS Education and Training Survey

◆ We’d like your feedback
  ◆ How do you use our software?
  ◆ What training do you need?
  ◆ How can we help you succeed?

◆ Free Virtual Campus Web Workshop
  ◆ Visit the Virtual Campus/Training/ESRI Press Island
  ◆ Online until August 15, 2005

◆ Web site: www.esri.com/trainingsurvey/
Thank you!

Questions?

Reminder: Workshop evaluation