ArcSDE Data Preparation Techniques Enhance Drawing Performance

By Mark Harley, Geographic Data Technology/Tele Atlas North America

GIS software is commonly used to generate maps for interactive applications. Speed is often critical in such situations. For example, people browsing the Web may lose interest and click elsewhere if they don't get a prompt response. In certain instances, it can be advantageous to precalculate and store selected information rather than requiring the GIS software to generate it on the fly for each map drawn. This article describes a few techniques that can be used to prepare data ahead of time to save drawing time later.

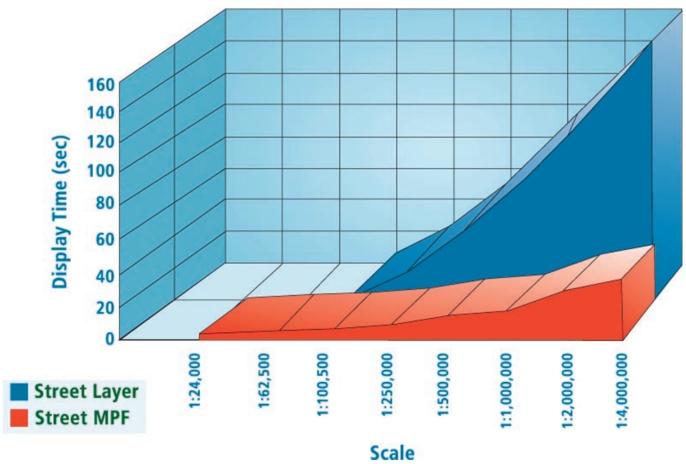

Figure 1: This chart shows the difference in ArcMap draw times between a regular street layer and a multipart feature street layer.

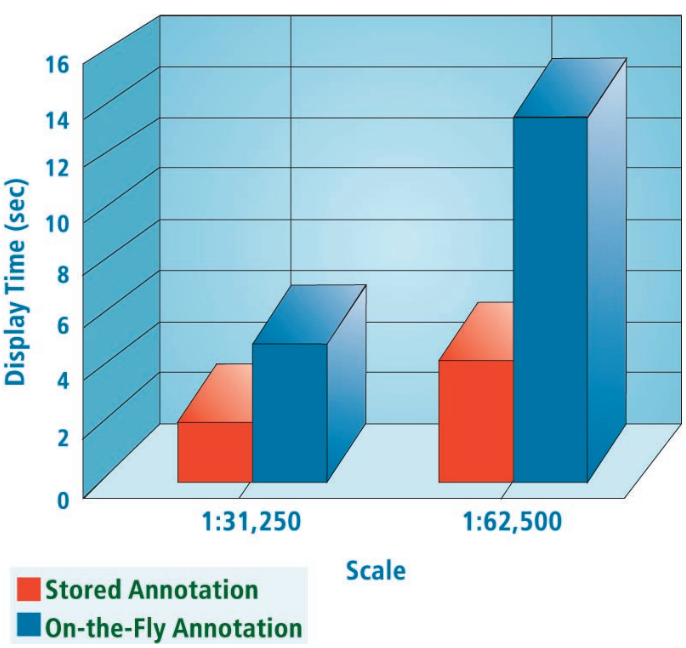

Figure 2: A comparison of ArcMap draw times for both precalculated annotation and labels drawn on the fly at two scales

When to Use Data Preparation Techniques

The first step in considering the use of data preparation techniques is to decide when, and even if, these techniques would be beneficial. In many cases, performance turns out to be adequate when using data in an unaltered form. If early tests suggest that performance will meet system requirements, no further work is necessary. While speed can probably be improved through the use of data preparation techniques or other system tuning, time may be better spent on other activities.

If initial benchmarks show that drawing may fail to meet system performance requirements, additional work should then proceed. Analysis should be performed to identify one or more potential data preparation techniques or tuning activities that will achieve the desired results. The following techniques may be used to improve system performance through data preparation, also known as preprocessing, preformatting, or nonreal-time processing.

Feature Grouping or Multipart Features

When a large number of features must be rendered frequently as part of a map image, feature grouping can be used to improve performance. Combining features reduces the number of individual rows that must be retrieved from the relational database used by ArcSDE.

First, a quick review of how ArcSDE typically stores features. In the Oracle and Microsoft SQL Server implementations, ArcSDE (in the case of SDEBINARY or SDEBLOB) uses a feature table (or F table) to store the geometry of a given feature. To draw a map showing the geometry only (that is, without attributes), ArcSDE queries its spatial index to obtain a list of identifiers corresponding to features present in the window of interest. Then the F table records are retrieved—one for each feature in that window. In a dense map, this can mean accessing a very large number of rows in the database, which sometimes results in slower than desired performance.

Processing the data ahead of time into grouped or multipart features can reduce the number of database accesses and improve performance. By combining the geometry from many features into a single, much larger entity that contains all the geometric shapes of the individual features, only one corresponding row in the business table and one set of attributes are retrieved. Attributes common to the combined set of features can be retained, but values unique to individual features will be lost.

A detailed example may help describe this data preparation method more clearly. Consider a street network for a large metropolitan area. If each side of each city block in the entire region is stored as a separate line segment, there may be tens or hundreds of thousands of lines used to represent the street network. Drawing a map zoomed out to a large scale may be slow because each individual line must be retrieved. When drawing large sections of the metropolitan area as multipart features, only a few F table rows containing the geometry for the entire map are retrieved instead of thousands of database records,

Still working with the street network example, if the freeways are grouped into one set of multipart features, major arteries into another group, and neighborhood roads into a third group, each road type can be drawn using different line widths and/or colors that indicate its type. Although the individual attributes, such as street names, are not present in multipart features, the map image will correctly show the road classifications. Individual attributes, such as street names, can still be made available to an application by accessing the original source layer when necessary.

Figure 1 illustrates the difference in ArcMap draw times between a regular street layer and a multipart feature street layer. To perform the test, streets for the entire state of California were loaded into ArcSDE 8.3 using an Oracle database in two formats, both as individual line features and the same geometry stored as multipart line features. The multipart feature layer was created using an application written in C using the SDE API. Similar layers can also be created using ESRI's sdegroup command. [For information on using the sdegroup command, see "Enhancing ArcSDE Layer Display Speed" in the April–June 2003 issue of *ArcUser,* which is available online at www.esri.com/arcuser.] Each map was centered on downtown Los Angeles and zoomed to the scales shown on the *x*-axis of the chart.

Note that when zoomed in fairly tightly, ArcSDE is able to retrieve individual line features at about the same rate as the multipart line features representing the same area. As the map is zoomed out, however, the street layer draw time increases from a few seconds to several minutes while the multipart street layer renders dramatically faster.

Precalculated Annotation

ArcMap has the ability to calculate annotation on the fly as a map image is rendered. ArcMap can label text for one or more feature classes without obstructing other important information on the map. With certain applications, particularly when the display is dense and the map is zoomed out, the time required to generate the needed annotation might be lengthy and slow response time to an unacceptable level.

Alternatively, annotation can be generated ahead of time and stored in an ArcSDE layer. It can be saved as a text string that includes the font size and angle of rotation at which it must be displayed. To work effectively, each zoom scale at which the annotation will be displayed and all of the visible layers to be enabled must be determined ahead of time. This allows for advance calculation of label placement to select reasonable locations for the label text.

The primary advantage of this technique is that it saves time in rendering maps that contain a large amount of annotation. A disadvantage is that the annotation is preset only for certain scales and combinations of visible layers. Figure 2 shows a comparison between ArcMap draw times for both precalculated annotation and labels drawn on the fly at two scales. The precalculated annotation layers proved to be significantly faster than allowing ArcMap's labeling engine to generate labels for each feature in real time.

Data Generalization for Scale Dependent Drawing

Scale dependent drawing is a feature that has been available with ESRI software for many years. It involves selecting the appropriate layers to be mapped based on the current zoom scale. The technique of precalculating generalized data is straightforward and definitely worth considering. This is most useful when dealing with data that must be displayed at a wide range of scales.

Drawing state boundary polygons for the United States provides a good example of this concept. When zoomed in to the local level, highly detailed state boundary data should be used to show state borders. When initially drawing a map of the entire United States, the fine detail of the highly accurate layer would be lost and data retrieval will most likely be slow. An appropriate data preparation technique would be to precalculate several generalized state boundary layers and enable appropriately detailed data at different scales. The Generalize tool in the ArcMap Advanced Editing toolbar uses the Douglas–Poiker algorithm to simplify the geometry of a selected feature. This method could be used to thin out simple polygon features and create state boundary layers that contain an appropriate level of detail for drawing at several zoom scales.

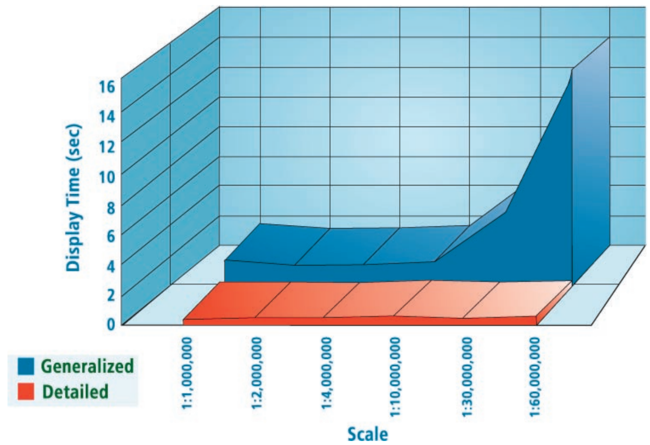

Figure 3: The difference in draw times between generalized and nongeneralized sets of state boundaries for the United States differ markedly at scales over 1:10,000,000.

Figure 3 shows the difference in draw times between generalized and detailed sets of state boundaries for the United States. Although some of the more detailed states contain several thousand points, the number of polygon features is not large. Notice that when zoomed in to a fairly small area, the performance boost is relatively small. When drawing the full outline for all states, the generalized map renders in a fraction of a second while the detailed version takes nearly fifteen seconds. In an interactive situation, it is clearly preferable to use the generalized boundary when zoomed way out.

Reprojecting Data

ArcSDE allows data to be stored using different coordinate reference systems or projections in each feature class. ArcMap can be used to reproject each data layer on the fly to the target projection for the overall rendered map image. In many situations, prior conversion is not necessary because reprojecting in real time will be sufficiently fast. However, sometimes preprocessing the data into the target coordinate system can save significant map drawing time. This data conversion can be performed as a feature class is loaded into ArcSDE and improves performance by paying for the reprojection once rather than requiring conversion each time the data is retrieved for drawing.

Database Denormalization

Database designers are trained to normalize data. They move repeating groups of items and independent fields to separate tables to avoid storing multiple copies of the same information. Although it is cleaner to organize and maintain information in properly normalized databases, a side effect of normalization is that more table joins may be required to satisfy certain queries.

If required queries perform acceptably with all data normalized, it makes sense to maintain the database in proper relational form. However,

Continued on page 40

ArcSDE Data Preparation Techniques Enhance Drawing Performance

Continued from page 39

if speed is a problem, selective denormalization may help. For example, when storing ZIP Code polygons, normal form would be to maintain the ZIP Code as an attribute directly on the business table and keep the associated post office names in a separate table. However, if the additional table joins required to select the primary name string for a given ZIP Code were slowing an important query significantly, the primary name string could be duplicated in the business table.

Summary

When designing a GIS application, rapid system response time is often critical, particularly when the application is interactive. Data preparation techniques can be a good way to improve system performance. The specific speed increase realized from using any particular data preparation method will vary depending on the computing environment; number of simultaneous users; the precise nature of the data; and, perhaps—most important—the way the data is typically accessed.

It is important to evaluate performance before investing time on data modifications. If the system performs well enough with the data in its normal format, up-front data preparation may not be warranted. It is worth mentioning that additional elements should be considered when designing for ArcSDE system performance. System-level tuning of the interactions between the operating system, relational database, ArcSDE, and the end user application is critical. Proper server sizing (i.e., adequate hardware selection) is also vital. For further information on server sizing, the ESRI white paper *System Design Strategies* (www.esri.com/library/whitepapers/pdfs/sysdesig.pdf) is highly recommended.

For more information, contact

Brad Gellerstedt, Senior Software Engineer
E-mail: brad_gellerstedt@gdt1.com
Mark Harley, Director, Consultative Services
E-mail: mark_harley@gdt1.com

Geographic Data Technology/Tele Atlas North America
11 Lafayette Street
Lebanon, New Hampshire 03766-1445
Tel.: 603-643-0330
Fax: 603-653-0249
Web: www.geographic.com and www.teleatlas.com