P-121

SCALABILITY OF CONTEXTUAL GENERALIZATION PROCESSING USING PARTITIONING AND PARALLELIZATION

BRIAT M.O., MONNOT J.L., PUNT E.
ESRI, REDLANDS, UNITED STATES

Contextual generalization tools need to efficiently analyze spatially related data. The generalization tools in ArcGIS 10 use either a TIN structure or an optimization engine which both rely on in-memory data structures. These tools are able to process efficiently approximately 100,000 features, but run out of memory as the number of features increases towards datasets containing millions. Workflows can be created to overcome those limits, but there is a significant cost in terms of data management.

The approach we have been considering is to subdivide the dataset into partitions defined as polygons with a smaller geographic extent. This allows control over the number of features to process within each partition and is easily obtained by deriving rectangles from a quad tree structure.

The main challenge with this approach is to ensure a seamless processing that will be independent of the partition boundaries. A second challenge is to ensure that global processing is independent of the partition's processing order, which adds some constraint on the algorithms implementation.

Partitions are processed using a buffer zone that is derived from each tool's logic and will ensure that features inside the partition consider their relevant environment, and when applied to both sides of a partition boundary, ensures consistent processing at boundaries. This is however not necessarily true with tools based on optimization techniques since by design they may produce different valid results. In these cases such tools can use a locking mechanism to consider what has already been processed. In future releases of ArcGIS it is anticipated that geoprocessing tools will handle this problem, depending on their output type (new output, in place editing with attribute and/or geometry change).

With this approach, the contextual generalization tools are now scalable as there is no limit for the number of partitions. Since we control the maximum number of features per partitions, processing the entire dataset happens in linear time.

Non-adjacent partitions have no strong interaction, which makes it possible to look at concurrent processing. We will review how strong the adjacency constraint is whether the tool modifies data only inside the partition or also in a limited area outside the partition. We will look at the database access problem if multiple processes work on the same dataset and describe an implementation using multiple services based on the ArcGIS file geodatabase.

These techniques have been used successfully on average quad core computers to process state level datasets (several million features) from scales of 1:18k up to 1:1M. Using this approach we have produced several multi-scale Web maps. The usage of concurrent processing allows for reasonable processing time, no more than a couple of hours per million features and per scale.

ArcGIS has a database replication mechanism that can be used to distribute this processing work even more and overcome database access limitations.