Computing in the Cloud (CiC):
GIS Vector Data Overlay Computation
on Windows Azure Platform

Sushil Prasad           Xuan Shi

Georgia State University
NSF
Microsoft
Research Challenge

• How to improve the performance of vector overlay computation over large scale spatial data by utilizing Windows Azure Cloud platform?
Spatial Computation in the Cloud

Task(s) accomplished in a single desktop/standalone GIS
Concepts in Windows Azure Cloud

Web Role(s)

Worker Role(s)
Processing single files

- Dispatch
- Monitor
- Aggregate

Reprojection, create index, build pyramid, etc.
Raster data modeling

Partition/Dispatch

Monitor

Aggregate
Vector overlay computation

equal, touch, contain, within, intersect, difference, union, etc.

Partition/Dispatch

Monitor

Aggregate

Oops!

Help?

How?
Partitioning two sets of data

- Partitioning binary streams
  - Where to cut???

- Partitioning based on the order of input features
  - Within a layer, the order of input is meaningless
  - Between layers, the random orders generate more chaos
Uniform grid vs. tiled processing

- Split [sequential?] – compute [parallel] – merge [sequential?]
- Smaller cells vs. more overhead
- Load balance, monitor mechanism, etc.
Partitioning upon spatial index

- Spatial data have build-in spatial index [R-tree, Quad-tree, etc.]
- No APIs to manipulate data based on spatial index
- Building spatial index over two large scale datasets for data partitioning in Web role is time consuming
Partitioning vs. spatial relationship

- Data partitioning is determined by the *potential* relationship, i.e. the bounding box relationship
- Overlay computation determines the *true* spatial relationship
- No silver bullet for *all* kinds of spatial relationships
Data preparation and I/O streaming

• Computing nodes in cloud/grid/GPU may not be able to utilize proprietary modules
  - **Shapefile or spatial database:** looping through 500,000+ features to partition two datasets into cloud seems another process of spatial overlay computation
  - **GML:** before read through the whole file, nobody knows 1) how many features is has; 2) for each feature, what the bounding box is; 3) for each feature, whether it is a multi-polygon; 4) how many holes each exterior ring has; 5) how many vertices each ring has
  - **New data schema** designed to enable efficient data partitioning and processing
    - Stored in Azure tables
The general workflow

Web Role

1. Parse XML and store as objects defined in the new schema
2. Sort Polygons in parallel based on bounding boxes
3. Link Base Layer to Overlay Layer
4. Serialize and store into Azure table
5. Add messages to work queue for each job
6. Wait for Output Queue to be populated
7. De-serialize and write to output file

Worker Role

1. Wait for work queue to be populated
2. Read from table and de-serialize
3. Feed Polygon to GPC library
4. Serialize and store the output into Table
5. Populate output queue
Processing in the cloud
Aggregation

• Aggregation may be simplified in case of intersect, touch, contain, within operations – the Web roles only collects and write out the results without any further processing.

• Aggregation can be a challenge in other spatial operations, such as union, which may need a different partitioning solution
Project under development

Questions?