Big Data Geo-Analytical Tool Development for Spatial Analysis Uncertainty Visualization and Quantification Needs

Baker, D.V. "Vic"[1], Bauer, Jennifer[2,3], Rose, Kelly[3]

[1]Mid-Atlantic Technology, Research & Innovation Center, South Charleston, West Virginia
[2]AECOM for the US DOE, National Energy Technology Laboratory, Albany, Oregon
[3]US Department of Energy, National Energy Technology Laboratory, Albany, Oregon

Abstract

As big data computing capabilities are increasingly paired with spatial analytical tools and approaches, there is a need to ensure uncertainty associated with the datasets used in these analyses is adequately incorporated and portrayed in results. Often the products of spatial analyses, big data and otherwise, are developed using discontinuous, sparse, and often point-driven data to represent continuous phenomena. Results from these analyses are generally presented without clear explanations of the uncertainty associated with the interpolated values. The Variable Grid Method (VGM) offers users with a flexible approach designed for application to a variety of analyses where users need to study, evaluate, and analyze spatial trends and patterns while maintaining connection to and communication of the uncertainty in underlying spatial datasets. The VGM outputs a simultaneous visualization representative of the spatial data analyses and quantification of underlying uncertainties, which can be calculated using data related to sample density, sample variance, interpolation error, uncertainty calculated from multiple simulations, etc. In this presentation we will show how we are utilizing Hadoop to store and perform spatial analysis through the development of custom Spark and MapReduce applications that incorporate ESRI Hadoop libraries. The team will present custom 'Big Data' geospatial applications that run on the Hadoop cluster and integrate with ESRI ArcMap with the team's probabilistic VGM approach. The VGM-Hadoop tool has been specially built as a multi-step MapReduce application running on the Hadoop cluster for the purpose of data reduction. This reduction is accomplished by generating multi-resolution, non-overlapping, attributed topology that is then further processed using ESRI's geostatistical analyst to convey a probabilistic model of a chosen study region. Finally, we will share our approach for implementation of data reduction and topology generation via custom multi-step Hadoop applications, performance benchmarking comparisons, and Hadoop-centric opportunities for greater parallelization of geospatial operations. The presentation includes examples of the approach being applied to a range of subsurface, geospatial studies (e.g. induced seismicity risk).

Goals

- **Develop Big Data GIS tools that leverage parallel processing and high performance computing**
- Use these capabilities to perform **data mining, analysis, simulation, and modeling** of high volume, velocity, and variety spatial and non-spatial data

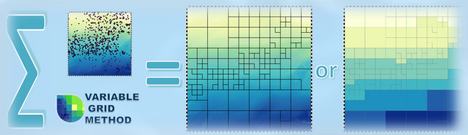
This poster illustrates one such Big Data GIS tool developed at NETL that utilizes a Hadoop cluster with ESRI-Hadoop libraries to implement a highly parallel and horizontally scalable grid generator for the Variable Grid Method.

Variable Grid Method (VGM) 101

The VGM is a spatio-temporal approach designed to more effectively communicate the underlying uncertainty associated with the analysis of big data (Bauer and Rose 2015). The VGM helps quantify and visualize spatial data and uncertainty simultaneously, using concepts of clarity & resolution to represent uncertainty so that smaller grid cell sizes represent areas with less uncertainty and as grid cell sizes increase so does the uncertainty associated within the grid cell.

VGM approach:

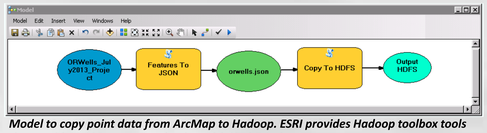
- allows the ***flexibility*** to use different data types and uncertainty quantifications
- preserves ***overall spatial trends and patterns*** observed within the data
- enables users to ***customize*** the final product to meet their needs and best communicate results in an ***intuitive manner***



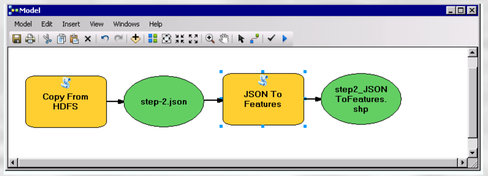
Communicate data (via colors) and uncertainty (via grid cell size)

Reference: Bauer, J.R., and Rose, K., 2015, Variable Grid Method: an Intuitive Approach for Simultaneously Quantifying and Visualizing Spatial Data and Uncertainty, *Transactions in GIS*. 19(3), p. 377-397

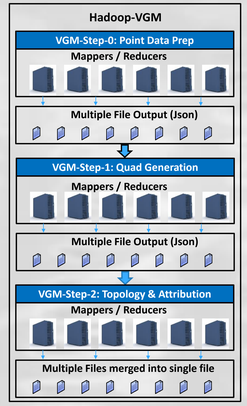
Hadoop-VGM: ArcMap and Hadoop Overview



Model to copy point data from ArcMap to Hadoop. ESRI provides Hadoop toolbox tools that enable ArcMap communication with Hadoop cluster.



Model to copy results from Hadoop into ArcMap



Hadoop-Based VGM Detailed Workflow

Well data from ArcMap


```
{"hasM": false,
"spatialReference": {"wkid":4326},
"features": [
{
"attributes": {
    "UWI__APINu": 3708520259.0,
    "OR_Base_m_": 0.0,
    "Surf_Lat": 41.484683,
    "Salinity__": 0.0,
    "WSN": 1.0,
    "Surf_Lon": -80.103193,
    "Brine_Dens": 0.0,
    "OR_Gross_T": 0.0,
    "Porosity__": 0.0,
    "NET_THICKN": 0.0,
    "Oriskany_T": 1190.549
},
"geometry": {
    "y": 5084098.520442805,
    "x": -8917047.03754837,
    "spatialReference": {
        "wkid": 4326,
        "latestWkid": 4326
    }
}
},
{
"attributes": {
    "UWI__APINu": 3703920665.0,
    "OR_Base_m_": 1977.9,
    "Surf_Lat": 41.730652,
```


Example JSON from ArcMap

VGM-Step-0

Description: Convert 'enclosed-Json' ESRI feature class into 'feature-per-row' unenclosed-Json.

Input: 'Enclosed-Json' formatted data (i.e., ORWells-wgs84.json) uploaded from ArcMap using ESRI/Hadoop toolbox tools 'Features to Json' & 'Copy to HDFS'.

Output: Processed 'Unenclosed-Json' with 'feature per row' layout suitable for Mapper.

Mapper (Setup): Create EsriFeatureClass from input file and write each feature as a row represented as unenclosed-Json.

Reducer: Aggregate Mapper output into one or more files

VGM-Step-1

Description: Generate multi-resolution bounding quads for input point data set (i.e., ORWells-wgs84)

Input: vgm-step-0 output 'Unenclosed-Json' of row-per-feature representation of orwells-wgs84 data

Output: Quads of varying extents with attribution (i.e., point count, max/min/avg salinity, porosity, brine density)

Mapper (Setup): Load point features from vgm-step-0 and use to generate quadtree node extents.

Mapper: Feed mapper each row of 'unenclosed-Json' from vgm-step-0 point data and query the quadtree for all quads that contain

Reducer: Aggregate Mapper output into one or more files and store in vgm/working/output-0/.

VGM-Step-2

Description: Generate non-overlapping topology of vgm-step-1 quads and calculate well point data per new geometries.

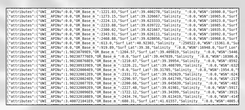
Input: Multi-resolution quads generated in vgm-step-1 AND the point data generated from vgm-step-0

Output: Non-overlapping polygons as 'unenclosed-Json' features with attribution (point count, min/max/avg porosity, etc.)

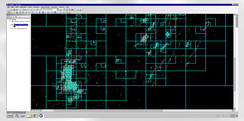
Mapper (Setup): Load vgm-step-1 output files representing attributed quads of varying resolutions to generate non-overlapping topology.

Mapper: Feed the Mapper with rows from the vgm-step-0 'unenclosed-Json' point feature data, query topology for 'point in polygon' to generate polygon's attributes, and perform geometry subtraction using ESRI Hadoop libs

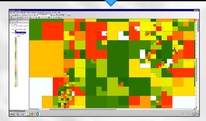
Reducer: Tally the attributes for each polygon and write attributed polygon as unenclosed-Json.



'Feature per row' formatted data for MapReduce



Overlapping attributed quads (shown via ArcMap)

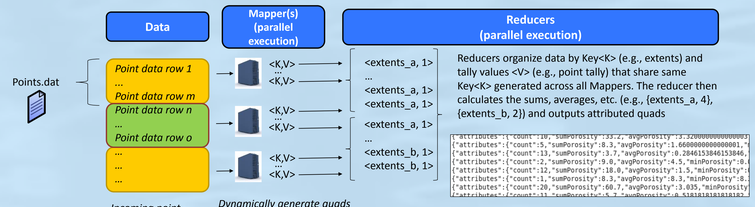


Attributed polygons for ArcMap

Merging GIS and Big Data computing for advanced 3D/4D geospatial analysis for uncertainty quantification and visualization allows researchers to:

- ***Offload intensive geometric operations*** from desktop to a Hadoop cluster
- Is ***highly scalable*** **-** immediately increasing compute power by adding additional nodes
- Is ***self healing*** in the event that node(s) go offline,
- The approach is ***ideal for executing parallel operations on geometric operations involving many features***. Furthermore, ***no ESRI license required*** – ESRI provides free Java-based Hadoop geometry and spatial libraries in addition to freely available ArcMap toolbox tools to connect and transfer data between ArcMap and Hadoop.

High Level: How Hadoop MapReduce Works -- Attributing Polygons in Parallel



Incoming point features that will be used to attribute polygons

Dynamically generate quads via quadtree and perform "point in polygon" where <K,V> = <quad_extents, 1> for each point within polygon


```
{"attributes":{"count":10,"sumPorosity":33.2,"avgPorosity":3.32000000000000},"
{"attributes":{"count":5,"sumPorosity":8.3,"avgPorosity":1.660000000000001,"
{"attributes":{"count":13,"sumPorosity":3.7,"avgPorosity":0.2846153846153846,"
{"attributes":{"count":2,"sumPorosity":9.0,"avgPorosity":4.5,"minPorosity":0.
{"attributes":{"count":12,"sumPorosity":18.0,"avgPorosity":1.5,"minPorosity":
{"attributes":{"count":1,"sumPorosity":0.3,"avgPorosity":0.3,"minPorosity":0.
{"attributes":{"count":20,"sumPorosity":60.7,"avgPorosity":3.035,"minPorosity
```


(Actual attributed polygons generated by Hadoop MapReduce approach represented in JSON format)

How Does MapReduce Work: Thought Example

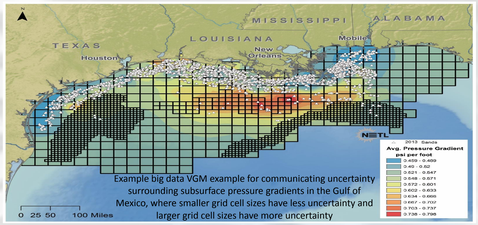
Example:
4 friends are playing cards. The cards spill on the floor. Pickup and **organize** (by suit) the spilled deck of cards

Solution:
Have each friend grab some of the cards and organize their cards by suit.
This is analogous to **'Mapping'** in Hadoop

Combine each friend's stacks of cards (organized by suit) and combine like suits. This is analogous to **'Reduce'** in Hadoop

Results to date:

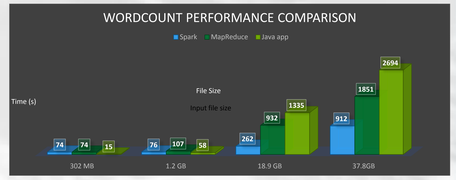
- Successfully implemented MapReduce based VGM-Hadoop prototype that generates non-overlapping, attributed geometries and integrates with ArcMap
- MapReduce implementation required creation of three unique Hadoop applications due to single-pass nature of MapReduce
- Approach has been tested with a data set consisting of 1 million points



Example big data VGM example for communicating uncertainty surrounding subsurface pressure gradients in the Gulf of Mexico, where smaller grid cell sizes have less uncertainty and larger grid cell sizes have more uncertainty

Benchmarking:

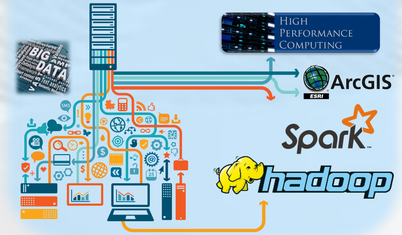
Spark vs MapReduce vs Single Threaded Application



We compared execution times for varying size data sets using Hadoop cluster-based MapReduce and Spark vs a stand alone, single threaded Java application (running on the Hadoop cluster's main node). The tests performed a "Word Count" algorithm to tally the occurrences of each word in the input file. The chart illustrates that as the data size increases, Hadoop-based applications (particularly Spark due to it's in-memory design) outperform the single-threaded Java application (smaller time values are better).

Next Steps:

- Implement Spark "in memory" version for iterative, multi-pass processing to improved geometry merge / dissolve options and bi-directional quadtree traversal
- Develop additional Big Data GIS tools
- Expand hardware capabilities to better meet data needs with cluster/cloud analysis and integration of Big Data and High Performance Computing techniques



Funding for this project is part of the US DOE Phase 1 SubTER crosscutting initiative

Technology Transfer

Variable Grid Method, U.S. 14/619,501. For information about this technology please see http://www.netl.doe.gov/business/tech-transfer/available-technologies/tech-details?id=88395c58-97a0-4dbf-87b6-880d3c4a915b