Spatiotemporal data representation and its effect on the performance of spatial analysis in a
cyberinfrastructure environment
- A case study with raster zonal analysis

Corresponding Author: Wenwen Li, Wenwen@asu.edu

Abstract:

This paper conducts a systematic research to uncover the impact of spatiotemporal data representation on the performance of raster analysis in a cyberinfrastructure environment. Two broad categories of data organization based on file system and database system are presented and discussed. In particular, these include five specific approaches of storing time-series raster data involving tiling (partitioning the entire image file into non-overlapping pieces), stacking (compositing multiple single-band images into a large multi-band image) techniques, and a combination of tiling and stacking in files or database tables. Raster zonal statistics, which have been used to support a variety of GIS applications ranging from watershed analysis to summarizing forest products, is selected as an example raster analysis algorithm. A series of experiments were conducted to evaluate the performance of the five proposed approaches using different spatial and spatiotemporal queries. The results show that spatiotemporal data representation, though largely ignored in the design of a cyberinfrastructure system, does play an important role in system performance. Specifically, tiling techniques with the support of spatial database outperforms all other approaches, especially those adopting stacking techniques in the data organization. For illustration, the best raster analysis solution was implemented and integrated into an operational cyberinfrastructure in the context of providing spatial decision support in polar science. We expect this work to offer insights to develop efficient cyberinfrastructure modules to support spatial analysis through a thorough analysis of spatiotemporal data representation.

Keywords: spatiotemporal data representation, polar cyberinfrastructure, performance, polar science, web service

1. Introduction

The demand for cyberinfrastructure to support spatial decision-making has been rapidly increasing. Geospatial Cyberinfrastructure (GCI), a domain-specific research area centered on geospatial data, seeks to provide better organization, high-performance analysis and visualization of georeferenced resources scattered in the cyberspace. Current applications of GCI, ranging from physical to social science domains, include cyberinfrastructure for field data collection and management (Hey et al., 2009), cyberinfrastructure-enabled science gateways (Goff et al., 2011; Moore et al., 2013), and cyberinfrastructure for high performance climate (Peters-Lidard et al., 2007), atmospheric (Droegemeier et al., 2004), ocean observation (Song et al., 2012), and/or urban economic modeling (Li et al., 2013). Development of key technology to advance cyberinfrastructure research is expected to become crucial in accelerating knowledge discovery in many science disciplines (NSF, 2009). In geospatial sciences, cyberinfrastructure research is focused primarily on developing parallel algorithms on high performance computing (HPC) platforms to improve performance of geoprocessing (Allcock et al., 2002; Joussaume, 2009; Laura et al. 2015; [Li et al. 2015](#); Song et al., 2014; Wang et al., 2005; Yang et al., 2008; Zhang and Tsou, 2009;). Notable work includes the NSF CyberGIS project led by Wang (2010) and the geospatial cloud computing initiative led by Yang et al. (2011a). The representation of spatial data, especially the representation of spatiotemporal data, is the foundation for many spatial analysis routines. However, it is largely ignored in cyberinfrastructure research.

Taking time-series raster data processing as an example, a commonly used operation is to conduct zonal analysis in order to obtain statistical information. This operation, also known as the zonal statistics analysis, has a widespread use in a variety of geospatial domains. For instance, raster zonal statistics has been employed to study the Urban Heat Island effect (Kardinal Jusuf et al., 2007; Rinner and Hussain, 2011). In Economics, zonal statistics is employed to summarize raster data of forest products, agricultural products, and tourist services to evaluate the regions' economic value (Chen et al., 2009). Raster zonal statistics is also a useful tool for evaluating the accuracy of DEM (Digital Elevation Model) data (Shan et al., 2003) [and](#) a critical method for summarizing raster values over [a](#) watershed to extract hydrological parameters (Whiteaker et al., 2007). Zonal statistics has also been used to improve the accuracy of land use and land cover (LULC) classification through combining images of different bands and eliminating mixed pixels (Saadat et al., 2011).

Despite its popularity the performance of raster zonal statistics, especially when dealing with big spatial and temporal data, remains a major concern (Mennis, 2010). Existing studies focus on the use of parallel computing to improve computational efficiency (Zhao et al., 2012; Zhang and Simin, 2013). The effect of raster data organization and representation on the performance of analysis has rarely been discussed in the literature. This issue however cannot be ignored in scientific analysis, since it is an essential aspect of the data preparation step which often takes the majority time in the whole knowledge discovery process.

Data representation and organization is also a key component in a geospatial cyberinfrastructure, which motivated this specific study (Jabeur et al., 2009; Yang et al., 2011b).

Our goal is to build a cyberinfrastructure for online analysis of the time-series polar (snow) data by investigating the effect of different data representation and organization approaches on the performance of raster analysis. The zonal statistics is taken as an example. The rest of the paper is organized as follows: Section 2 briefly reviews past work and commonly-used strategies and models for representing, organizing and storing time-series raster data. Section 3 presents two different solutions for zonal statistics analysis. Section 4 proposes five different strategies for storing and organizing raster data. Section 5 discusses the experimental result and identifies the best model to support raster analysis. Section 6 introduces the integration of the raster analysis algorithm in a polar cyberinfrastructure portal to support polar environmental analysis. Section 7 concludes the paper and proposes future research directions.

2. Background

2.1 Spatiotemporal data representation

Spatiotemporal data representation provides an opportunity to improve the performance of raster zonal statistics, because different raster data organization methods can result in different I/O consumption and data query concurrency, which strongly influence the performance of raster-based statistics. Conceptual models regarding the representation of spatiotemporal raster data include the snapshot model, the space-time composite model, the spatiotemporal object model (Saeed and Reza, 2003; Yuan, 1996).

The snapshot model represents the world as overlays of time-stamped layers. In this model, raster layers of different time stamps are stacked to represent the continuous changes in a scene (Armstrong, 1988). The Temporal Map Sets (TMS) is a typical snapshot model (Yang et al., 2005). The limitation of this model is that the temporal information, including timestamps and temporal intervals is represented only implicitly because they are usually recorded in an additional file (e.g. a metadata file, a header file) rather than being directly embedded in the data. Also, there is a lack of a standardized metadata to encode the temporal information for machine interpretation.

In a space-time composite model, temporal change information is composited with spatial features and is recorded as a field-based attribute (Segev and Shoshani, 1993). This method is more suitable for representing discrete objects, such as a moving object, than continuous raster data, which describes changing scenes/phenomena over a geographic extent.

For a spatiotemporal object model, a time dimension is added to the two-dimensional space and an object with a state is marked as a geo-atom (Goodchild et al., 2007). In this model, an object with spatiotemporal dimensions is represented as a geo-object (Goodchild et al., 2007). Since this geo-object model is abstracted from point sets, it is also challenging to represent time-series raster data.

2.2 Spatiotemporal data organization and storage

In terms of data organization, raster data can be structured and stored using a file-based model or a spatial database model. In environmental studies, summary statistics is applied on the biweekly-AVHRR (Advanced Very High Resolution Radiometer) to generate time-series of normalized difference vegetation index (NDVI) data (Eidenshink, 1992). These data are

composited and stored in a raster file system. Yuan (2001) develops a hierarchical model to represent time-series event for precipitation accumulation analysis. Another example can be found in Carroll et al.'s work (2009), in which 16-day worth of MODIS (Moderate-Resolution Imaging Spectroradiometer) data were composited into a file for water mask generation. Despite the advantages of easy data organization, the file-based system suffers from the consumption of very large memory space as well as lack of efficient ways to handle concurrent requests. In contrast, spatial database technique has become another promising solution for representing and storing raster data due to its high integrity and compatibility (Pelekis et al., 2004; Agosto, 2013). Zhou et al. (2009) conducted a benchmark study to compare the spatial query efficiency of various spatial databases. The representation of spatiotemporal raster data is not considered in the above studies.

In summary, existing studies focus more on developing new space-time models to represent the temporal change of a spatial object. The effect of spatial-temporal data model on the performance of data analysis in an operational environment has not been systematically explored. Next section introduces a typical raster data analysis method – the zonal statistics, as the major algorithm to be tested with the proposed spatial temporal data representation models.

3. Open-source solutions to support raster zonal statistics

Due to its popularity, raster zonal statistics has been supported in almost all mainstream GIS packages (Bunting et al., 2014; Cagnacci and Urbano, 2008; Clewley et al., 2013). Most of these tools, such as QGIS or GRASS, are designed for use in a standalone and single-user environment, which is unwieldy in supporting cyberinfrastructure applications. In terms of potential cyber-solutions, there have been two technical paths. One is to use the combination of PostGIS, GeoTools and GeoServer; PostGIS is used to store spatiotemporal data, GeoTools provides backbone support for zonal analysis, and GeoServer makes geospatial data and tools available as (Open Geospatial Consortium)-compliant Web Coverage Services (WCS) and Web Processing Services (WPS), respectively. Another solution is to develop coupled data and processing service – the OGC WCPS (Web Coverage Processing Service) to facilitate online zonal analysis. Rasdaman, an array analytics engine, supports for its implementation. In this paper, PostGIS and GeoTools are selected for conducting this analysis because they are open source and they are representative of two main data representation schemes (file-based and database approaches) in GCI. In addition, they support spatial filtering based on an arbitrary shape, such as the boundary of a state, which is often used in practical applications. Whereas the Rasdaman tool in its current version only supports a filter based on bounding box. Subsequent sections describe the implementation of zonal statistic algorithm in GeoTools (section 3.1) and a proposed solution in PostGIS (section 3.2) along with a discussion of methods to enhance performance.

3.1 Raster Zonal Statistics in GeoTools

The workflow of file-based zonal statistics in GeoTools is illustrated in Figure 1. First, raster data, e.g. in a GeoTiff file, and vector-format zones data, e.g. as an ESRI shape file, are loaded (step 1). Then the algorithm checks whether the bounding box of raster data intersects with the

Minimum Bounding Rectangle (MBR) of the polygon (step 2). For polygons intersecting with the raster data, a crop operation is performed on the raster (step 3) to extract the content of target extent from the image. Next, the **Region of Interest (ROI)** polygon is transformed from its original coordinate system to that of the target (step 4). Then, the location of each pixel is read out from the target image to determine whether it is within the ROI using a point-in-polygon test, and pixels falling outside the ROI will be filtered out. For an eligible pixel, its values on all required bands are scanned and grouped according to the ROIs. After all pixels are scanned, the sum, min, max, mean, and count are calculated for the associated values for each requested band.

Figure 1. The entire workflow of zonal statistics with the GeoTools open source package

There are two time-consuming sub-processes in this algorithm. One is the raster file I/O operation which needs to load target image data from the hard-drive into the system memory and consumes much time, especially when the raster to analyze has multiple bands. Another time constraint is that the Point-in-ROI operation (Step 5 in Figure 1) needs to be repeatedly invoked to obtain the locations falling inside the same ROI. To speed up this process, we developed a hash-table to store the intersected position on each grid line of the raster data with the boundary of the ROI. This hash-table has the ROI as the key, and the pairs of positions indicating which gridline segments fall inside the ROI as the value. When the same ROI is given, there is no need to perform the Point-in-ROI again. Instead, the pixel values falling inside the ROI can be directly collected and analyzed by looking up the intersection information stored in the hash-table. This improvement is especially useful to conduct the zonal statistics using the time-series data because the n (number of time slices) time-consuming Point-in-ROI operations

will only be required to execute once, therefore the overall performance can be significantly improved. The experiments in Section 5.1 demonstrate the performance improvement with this optimization strategy.

3.2 Raster zonal statistics based on spatial operators in PostGIS

The second solution for raster zonal statistics uses spatial database to store input dataset and relies on the build-in spatial operators in the database to conduct raster zonal statistics. It has different implementation from that of GeoTools. First, the data is imported to tables of the PostGIS database management system (DBMS), with a RASTER column storing the raster slice, and a GEOMETRY column storing the coordinates of polygon boundaries. Second, algorithm-wise, the ROI polygon is rasterized instead of being transformed into the coordinate reference system of the raster data. Figure 2 demonstrates the workflow of the database-based zonal statistics algorithm. The first step is to identify the possible intersection between the raster data and the ROI zone (polygon geometry) data. Next the clip operation is performed on the data records to extract the portion of the raster falling in the zone. As discussed, to conduct this spatial operation, the polygon zone data is converted to a binary raster data with the same geo-referencing information of the target raster data. A cell is assigned a value of 1 if it is covered by the ROI polygon, and NA otherwise. Cell-wise multiplication is performed on the target raster and the ROI raster data. This step results in a raster block with non-zero attribute values only in areas inside the ROI. Lastly, the summary statistics is executed on the new raster block to get the count, sum, average, min, and max value.

Figure 2. The algorithm flow of PostGIS zonal statistics

Figure 3 demonstrates two implementations of this algorithm for comparing the total snow volume in the states of Wisconsin and Colorado over any given period of time. In PostgreSQL statement 1 (Figure 3), multi-level nested sub-queries are used to combine three steps (namely, intersection, clip and summary statistics) into one query statement. Built-in spatial operators, including *ST_Intersects*, *ST_Clip* and *ST_SummaryStats* are invoked to execute the three steps. In PostgreSQL statement 2, this nested query structure is decomposed into individual queries. An inner-join based table is created in the first step to combine raster and zone layers that have intersections before the clip operation is performed to obtain the actual intersected parts. All intermediate results from each sub-query are stored in the temporary tables.

Experiment shows that the PostgreSQL statement 2 generates a 4-10 time speed up compared to PostgreSQL statement 1. This performance improvement is due to that (1) the repeated scan of the main data table was avoided by utilizing temporary tables to store intermediate results; and (2) the I/O congestion caused by "shared lock" and "update lock" during the concurrent access of raster and vector tables is also alleviated.

PgSQL Statement 1 For Raster Zonal Statistics

```
SELECT gid, (stats).*, state_name, fdate
FROM (
  SELECT gid, the_geom,
  --Step 3: Summary statistics
  ST_SummaryStats(
  --Step 2: Clip raster by a polygon geometry
  ST_Clip(rast,1,the_geom,true)
  ) AS stats,
  state_name, fdate
FROM raster_table
--Step 1: Intersection determination
INNER JOIN (
  SELECT gid, the_geom, state_name
FROM states)
AS vector_table
ON
  state_name IN ('Wisconsin','Colorado') AND
  ST_Intersects(the_geom,rast) AND
  raster_table.fdate >= '2014-03-30' AND
  raster_table.fdate <= '2014-03-31'
) AS query_results
```

PgSQL Statement 2 For Raster Zonal Statistics

```
Drop Table
IF EXISTS raster_geom_combination, cropped_raster;
--Step 1: Intersection determination
CREATE TABLE raster_geom_combination AS (
  SELECT gid, the_geom, state_name, rast, fdate
FROM raster_table INNER JOIN states
ON state_name IN ('Wisconsin','Colorado') AND
  raster_table.fdate >= '2014-03-30'
AND
  raster_table.fdate <= '2014-03-31' AND
  ST_Intersects(the_geom,rast)
);
--Step 2: Clip raster by a polygon geometry
CREATE TABLE cropped_raster AS
(SELECT gid, the_geom,
  ST_Clip(rast,1,the_geom,NULL,TRUE) AS stats,
  state_name, fdate
FROM raster_geom_combination
);
--Step 3: Summary statistics
SELECT (stats_result).*, state_name, fdate
FROM (
  SELECT ST_SummaryStats(stats) AS stats_result,
  state_name, fdate
From cropped_raster ) as query_results
```

Figure 3. Two kinds of implementation of raster zonal statistics using PostgreSQL/PostGIS

4. Strategy of raster data organizations to support efficient spatial-temporal analysis in GCI

In this section, we investigate two types of data organization methods (i.e. the file-based or the database-based methods) for representing time-series raster dataset in support of efficient spatiotemporal raster analysis. Five different data structures are investigated. For the file-based

data organization, time-series raster data is stored as a set of single-band raster data, with each image representing one time-slice (method f1). The second way is to consider each time slice as a band and store the time-series data as a multi-band raster file (method f2). For a database approach, tiling and stacking techniques are investigated. Tiling is to cut a raster dataset into a set of pieces for easy management and less memory consumption. Stacking is similar to the composition approach used in a file-based system. Approaches combining tiling and stacking are tested, including un tiled, unstacked data organization (method d1), a tiled, unstacked dataset (method d2), and a tiled and stacked dataset (method d3). Table 1 lists the short name for each method used to represent spatiotemporal raster data.

Method	Category	Description
f1	File-system	Multiple single-band raster files
f2		Single raster file with multiple bands
d1	Database	Untiled, unstacked raster in PostGIS
d2		Tiled, unstacked raster in PostGIS
d3		Tiled and stacked raster in PostGIS

Table 1. Different data representation methods

4.1 Raster data organization based on file-based strategies

For file-based data representation, GeoTiff(Ritter and Ruth, 1997) is selected as the primary image format to store raster data due to its popularity and widespread use in GIS applications (Gorokhovich and Voustianiouk, 2006; Moreno-Sanchez et al., 2007).

4.1.1 Method f1: Store time-series raster data in a set of standalone GeoTiff files

Figure 4 illustrates the data organization in method f1, which stores the data at different timestamps as individual GeoTiff files. These files follow a naming convention for quick data look-up. Taking the below daily snow depth data for example, each file starts with a data theme followed by year (YYYY), month (MM), and day (DD). Once a request is given, the temporal constraint is parsed and the corresponding raster files satisfying the constraint are selected. Then GeoTools picks eligible raster files, performs crop operation according to the algorithm described in Figure 1.

Figure 4. Using standalone GeoTiff files to store time-series raster data

4.1.2 Method f2: Merge time-series GeoTiff Files into a composited multi-timestamp GeoTiff File

In this strategy, the set of standalone raster data in Figure 4 is merged together into a single GeoTiff file named Multi-Timestamp Composited GeoTiff File (MTCGeoF), in which data at one timestamp is stored as a band, as illustrated in Figure 5(a). Utilizing the MTCGeoF to organize time-series data requires mapping a timestamp to a band number. Figure 5(b) illustrates a time metadata used for translating timestamps to band numbers. Two elements, “TimeCoverage” and “TimeSchema” are defined. “TimeCoverage” defines the start (T_1) and end dates (T_2) of a MTCGeoF, and “TimeSchema” describes the sampling interval (i) between bands. The band number n representing a specific time point in the MTCGeoF file can be computed as:

$$n = \text{floor} \left[\frac{\text{DateDifference}(T_1, T_2)}{i} \right]$$

The MTCGeoF approach belongs to the category of snapshot model by integrating the temporal metadata into the dataset to facilitate look-up of individual band data. When the time span is long, the volume of a raster file will become very large. The large amount of data will generate heavy processing load for a server due to limited capacity of system memory. Therefore, this approach is more memory intensive than the unstacked approach.

Figure 5. Illustration of MTCGeoF with temporal metadata

4.2 Raster data organization strategies based on raster-supported spatial database

Besides using the file systems to organize raster coverage, another approach is to store and manage raster data in raster-supported spatial databases. For a set of raster data in a time sequence, a scalable way to support temporal information in the raster data is to add an additional column with the type of DATE to store the corresponding timestamp to which the current tile belongs (Figure 6 and Figure 7). In this way, a raster layer with spatiotemporal information in the database is composed of a set of records or tuples with two required columns, RASTER and DATE (Figure 6 and Figure 7). An alternative way is to directly import a

multi-timestamp composited raster file (e.g. MTCGeoF) into a raster table with no DATE column needed (Figure 8), and use time metadata defined in section 4.1 to indicate temporal information. Below we discuss the combined tiling and stacking mechanisms for raster data storage in a spatial database.

4.2.1 Method d1: Storing untilted time-series raster data in a database

In this model, each time slice of temporal raster data is stored as a record in a data table. A column with a raster data type is utilized to store raster and a separate column of DATE type is used to store the timestamp of this raster. In this strategy, a new time slice of raster data is stored as a complete entity without being cut into sub-tiles. As shown in Figure 6, the value of the DATE column in each record is unique, and raster coverage is stored as the RASTER type which is a specific data structure for raster data in the PostGIS Raster extension (Iacovella, 2012). This model can be considered as a variant of the space-time composite model, in which the spatial information is stored as a RASTER data type rather than a vector object in each row of the table.

Figure 6. The illustration of storing untilted raster data in spatial database

4.2.2 Method d2: Raster data organization with unstacked tiles stored in a database

Figure 7 demonstrates a tiled, un-stacked raster data model. Different from the data model described in Figure 6, the raster data of each time slice is cut into non-overlapping tiles. Each small tile is stored as a record in the data table and labeled with a unique ID. For instance, if the tile size is 128*128, that means a single image of size 6935*3351 will be cut into 1485 tiles. For quick allocation of data in the region/tiles of interest and for effective temporal filtering, a spatial index can be created on the raster column according to the convex hull of each raster tile, and a time index for temporal querying can be created on the DATE column as well. In the PostGIS database, GiST(Blasby, 2001;Lopes Siqueira et al., 2009) is recommended for spatial indexing and B-tree is used for temporal indexing by default. The spatiotemporal indexing

greatly increases the speed of data lookup and enables sub-linear time searching and serves as a performance booster especially when many raster tiles are involved in a query.

Figure 7. The raster data organization with unstacked tiles stored in a spatial database

4.2.3 Method d3: Raster data organization in the format of stacked tiles stored in database

In the stacked tiles model, a set of raster data in a time sequence are merged into MTCGeoF before being imported into the spatial database. Then, the MTCGeoF is cut into tiles and is inserted one tile per table row. Each tile has several layers and each layer is associated with a time point. Such organization of raster data is illustrated in Figure 8. In this data organization method, there is no need to use a standalone DATE column to identify the timestamp, instead this information can be obtained from the layer/band number. The relationship between time points and band numbers can be established using the time metadata described in the section 4.1.2.

Figure 8. The raster data organization in the format of stacked tiles stored in a spatial database

5. Experiments

This section conducts a series of experiments to evaluate the performance of zonal statistics algorithms with two open source solutions (GeoTools and PostgreSQL) in a cyberinfrastructure environment by applying the five proposed models. The sample raster data used in the experiment is daily snow depth data of continental United States. The specification of this dataset is listed in Table 2:

Parameters	Value
Source	National Snow and Ice Data Center
Spatial Extent	[24.95042,-124.73375, 52.8754, -66.942083]
Temporal Extent	01/01/2014-03/31/2014
Cell size (unit degree)	0.0083333333*0.0083333333
Single image size	6939*3351 $\approx$ 22.0 Megapixels
Image size of time-series	6939*3351*t (for data of 64 days, it is 6939*3351*64 $\approx$ 1.38 Gigapixels)

Table 2. Parameters of raster data used in the experiments

5.1 Comparison of the performance of raster zonal statistics with and without the optimization strategy using GeoTools

This experiment is conducted to evaluate the performance improvement of zonal statistics using the optimized GeoTools solution. In this experiment, the spatial extent is set to be 4 states, and the temporal extent increases from 1 day to 64 days. The f1 model (single GeoTiff) is selected to conduct the experiments. As Figure 9(a) shows, the optimized GeoTools yields much better performance than the original implementation. As the requested days of data increases, this performance advantage become more prominent, reflected by the increase in the speedup curve shown in Figure 9(b). The improvement is contributed by caching intersected vertices, described in Section 3.1.

(a) Comparison of actual processing time (b) Speedup of processing time after optimization

Figure 9. Comparison of processing time of raster zonal statistics in the original and optimized GeoTools with increasing temporal extents

5.2 Performance comparison for the spatial database solution with and without query optimization

To illustrate the performance improvement in the database solution with the optimized spatial query settings, two sets of experiments were conducted. The first set of experiments fixed the temporal extent as one day for the requested data and recorded the performance difference with increasing spatial extent (the number of states). The second one fixed the requested spatial extent to be 4 states and compared the performance with and without optimization with increasing temporal extent in the unit of days. Figure 10 (a) and (b) demonstrate the results for the two experiments respectively. As seen, the processing time for zonal statistical analysis is substantially reduced after adopting the proposed query optimization strategy described in Section 3.2. The average speed up is approximately 4 in both cases. Note that the most effective data representation model d2 (unstacked, tiled raster data) is used in these experiments. When a less effective data model, such as stacked raster data, is adopted, the highest speedup for this optimized PostGIS solution can reach up to 10 times.

(a) Processing time for zonal statistical analysis with increasing spatial extent

(b) Processing time for zonal statistical analysis with increasing temporal extent

Figure 10. The effect of spatial query optimization to the performance of the PostGIS solution

5.3 Impact of runtime performance of zonal statistics analysis by varying spatial extents

Here, we investigate the effect of combinations of data storage and organization methods on the performance of zonal statistics analysis with optimized GeoTools and PostGIS. Solutions include (1) GeoTools with method d1, (2) PostGIS with method d1, (3) PostGIS with method d2 and (4) PostGIS with method d3 are explored. In this experiment, we fix the temporal scope and increase requested data amount through increasing spatial extent. The spatial extent refers to the number of states whose geometries are used as Regions of Interest (ROI) to crop the raster data. The spatial extent sare set to be 1, 2, 4, 8, and 16 states respectively.

Figure 11 demonstrates the processing time using the above solution techniques. The performance of zonal analysis for model f1 based on a file system and a spatial database (model d1) is very close. This means that the two algorithm implementations obtain similar performance. In comparison, the execution time on unstacked tiles (model d2) is much faster than that on untiled raster data (model d1 or model f1). This is due to the fact that the intersection check in the tiling solution to test whether a tile falls in the ROI can help remove areas falling outside of the ROI and therefore save time on the actual cropping operation. Moreover, the processing time for the stacked tiles approach based on a database (model d3) is the longest among all tested approaches because stacking method costs more time in data loading. The results for model f2 are not included in the figure because it has been measured that this model takes too long for the system to respond even when only one state's data is requested.

Figure 12 demonstrates the speedup of three database solutions, d1, d2, and d3, relative to the file system solution f1. Model d2 has the highest speedup, with 2-2.5 times as fast as using approach f1. Approach f1 and d1 is approximately the same in terms of runtime performance. The approach d3 is only half as fast as the unstacked untiled approaches (f1 and d1) and one-fourth as fast as the unstacked, tiling approach (d2) on a spatial database.

Figure 11. The processing time of zonal statistics with changing the number of states to crop the raster

Figure 12. Speedup of spatial database method (d1, d2, d3) comparing to the file-based method f1

5.4 Performance of raster zonal statistics to handle time-series data

In this experiment, we fix the spatial extent to be 4 states and increase the data size by increasing the time scope for zonal statistics. The execution time of five methods, including f1, f2, d1, d2 and d3 are recorded in Figure 13.

It can be observed that the runtime performance did not get much improved for querying time-series data when packing these data into a multi-band raster data, especially on a file system (method f2). Among all solutions, method f2 gives the worst performance. This is

6 | because when it takes extra time for the stacking method to load the entire dataset on a file
7 | system, affecting negatively the runtime performance. Though the same stacking technique is
8 | used in d3, its performance is much better than f2, primarily because the tiling strategy is
9 | introduced. For the unstacking approaches, f1 and d1 achieves similar runtime performance for
10 | handling time-series data.

7 |
10 | Figure 13. The processing time of zonal statistics with the change of temporal interval. The y
11 | value for f2 when requesting 64-days of data is missing because the data size (1.38 Gigapixels)
12 | exceeds the processing ability of GeoTools.

21 | The approach d2 continues to demonstrate the best performance over all approaches. Figure 14
22 | presents the speedup of the database solution d2, compared to method f1, f2, d1, and d3. When
23 | the temporal extent is small, the speedup of d2 can reach up to almost 20 times with respect to
24 | method f2. This advantage gradually diminishes when the requested temporal extent increases.
25 | This is because as the requested temporal bands increase, more bands of composited data need
26 | to be loaded into memory for the processing. Therefore, more time is wasted in the I/O part, so
27 | a smaller performance difference is observed. However, when applying stacking technique in a
28 | database environment (method d3), as the requested temporal data increases, the performance
29 | gap between d3 and the best-performed method d2 reduces significantly. That means, the
30 | database management system is better than the file system at memory management when large
31 | amount of stacked raster needs to be loaded.

Figure 14. Speedup of the model d2 to all rest models with increasing temporal extent

5.5 Performance evaluation of optimized approaches in handling big data

We further evaluated the performance of raster zonal statistics in handling big data using both optimized GeoTools and PostGIS methods. We resampled the daily snow depth data to generate a new dataset with the size equaling to 7*7 times of original raster data. For a raster data at one single timestamp, there contains 1.06 Gigapixels. In this experiment, the enhanced PostGIS with the data model d2 and the optimized GeoTools with Single GeoTiff model (f1) are used, because these two models have been proved to perform the best among all methods. We fixed the time period to be one day and changed the number of states from 1 to 16 to record the process time. From Figure 15, it can be observed that both solutions have the capacity of handling big data. When data from more states are requested, the database approach runs much faster than the file-based approach, demonstrating the good performance of the tiled data model with spatial database support.

Figure 15. Processing time of raster zonal statistics to handle big data using optimized GeoTools and PostGIS

In summary, PostGIS with unstacked tiled raster model (d2), which can also be considered as an extension to the snapshot model performs the best among all tested models for processing time-series raster data. The better performance is reflected by a shorter processing time and a better linear increasing trend when data volume is increased spatially and temporally. This approach also shows great scalability in handling big spatial data. The space-time composite model and its extensions (model f2 and d3), in general, do not perform as good in the real-time spatiotemporal question answering scenarios.

6. Integrating the data model into a geospatial cyberinfrastructure portal

The best-performed d2 model is implemented as an OGC WPS and integrated into an operational GCI portal to support online real-time raster analysis. GeoServer serves as the web service engine to wrap geoprocessing modules into a standard WPS. Both the raster and vector data needed for the processing are stored in PostGIS database and can be delivered across a network. The application is the snow volume computation from a snow depth data available on a daily basis. The snow depth data comes from Snow Data Assimilation System (SNODAS) data products of National Snow and Ice Data Center (NSIDC; 2004).

Figure 16 demonstrates the user interface of the GCI portal integrating online OGC data services from remote servers. The current screenshot displays the snow depth data from a WMS. The popup windows display the process to invoke the WPS service of zonal statistics and present the statistical results. The lower right window allows users to select the data theme (i.e. snow depth), and spatial and temporal extent for the zonal analysis. The spatial extent can take one or multiple placenames as input. Once received, the GCI portal can parse the placename and identify the corresponding vector zone data from its available WFSs. After receiving the region and time of interest, the GCI invokes the zonal statistics service and feeds the raster and vector data into it. The summarized results – the snow volume, for each state on a daily basis is returned and displayed in the statistical chart (lower left window in the screenshot). By integrating this effective raster data organization model, the GCI portal is capable of supporting efficient online spatiotemporal analysis for snow scientists.

Figure 16. A screenshot of a GCI portal integrating the zonal statistics service

7. Conclusion

In this paper, we have systematically studied data organization issues in the context of real-time spatiotemporal raster analysis. This research originates from both [geospatial cyberinfrastructure research and spatiotemporal data modeling to study](#) the effect of spatiotemporal representation of raster data on the runtime performance within a cyberinfrastructure framework. We discussed an extended version of snapshot and space-time composite models respectively and implemented them on both the file-based and database systems. Systematic experiments showed that the tiled snapshot model on a spatial relational database (model d2) outperforms all other models in terms of answering spatial and temporal questions. We have successfully integrated this model into a cyberinfrastructure portal to support real-time raster data analysis to support polar data analysis.

This work has made an important contribution to the body of knowledge of geospatial cyberinfrastructure research in terms of data modeling and representation of spatiotemporal raster data. In the future, we will extend our research in the following directions: first, we will adopt internal-tiling strategy (Baumann et al. 2014) to extend the file-based GeoToolssolution to accept tiled GeoTiff files for efficient raster analysis. We will also exploit whether file formats (e.g. BIP, BIL, BSQ) may become a factor that impacts the system performance. Besides, we will also compare the performance of our proposed solution framework, which integrates OGC WFS, WCS and WPS into an integrated analysis, with the solution enabled by OGC [WCPS](#)(Baumann 2010). Second, we will conjunct the proposed raster data model with a variety of raster analysis algorithms, in addition to the raster zonal statistics discussed in this paper. Third, we will investigate the spatiotemporal models for vector data, to further improve the performance for online spatial analysis in a cyberinfrastructure environmentto support different science domains, such as polar science (Li et al. 2014).

Acknowledgement

This work is supported by National Science Foundation (PLR-1349259; BCS-1455349; PLR-1504432), the Open Geospatial Consortium, and Key Laboratory of Geological Information Technology, China Ministry of Land and Resources.

Reference

1. Agosto, E., 2013. Vector–raster server-side analysis: a PostGIS benchmark. *Applied Geomatics* 5, 177-184.
2. Allcock, B., Bester, J., Bresnahan, J., Chervenak, A.L., Foster, I., Kesselman, C., Meder, S., Nefedova, V., Quesnel, D., Tuecke, S., 2002. Data management and transfer in high-performance computational grid environments. *Parallel Computing* 28, 749-771.
3. Baumann, P. (2010). The OGC web coverage processing service (WCPS) standard. *Geoinformatica*, 14(4), 447-479.
4. Baumann, P., Mazzetti, P., Ungar, J., Barbera, R., Barboni, D., Beccati, A., ... & Wagner, S. (2014). Big data analytics for earth sciences: the earth server approach. *International Journal of Digital Earth*, (just-accepted), 1-50.
5. Blasby, D., 2001. Building a spatial database in postgresql. Open Source Database Summit.
6. Bunting, P., Clewley, D., Lucas, R.M., Gillingham, S., 2014. The Remote Sensing and GIS Software Library (RSGISLib). *Computers & Geosciences* 62, 216-226.
7. Cagnacci, F., Urbano, F., 2008. Managing wildlife: A spatial information system for GPS collars data. *Environmental Modelling & Software* 23, 957-959.
8. Center, N.O.H.R.S., 2004. Snow Data Assimilation System (SNODAS) Data Products at NSIDC. Colorado USA: National Snow and Ice Data Center.
9. Chen, N., Li, H., Wang, L., 2009. A GIS-based approach for mapping direct use value of ecosystem services at a county scale: Management implications. *Ecological Economics* 68, 2768-2776.
10. Clewley, D., Whitcomb, J., Bunting, P., Moghaddam, M., Chapman, B., McDonald, K., 2013. Quantifying Uncertainty in Wetland Change Mapping for Alaska: Part I-Uncertainty in Input SAR Data, AGU Fall Meeting Abstracts, p. 0453.
11. Droegemeier, K.K., Chandrasekar, V., Clark, R., Gannon, D., Graves, S., Joseph, E., Ramamurthy, M., Wilhelmson, R., Brewster, K., Domenico, B., 2004. Linked environments for atmospheric discovery (LEAD): A cyberinfrastructure for mesoscale meteorology research and education, 20th Conf. on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology.
12. Eidenshink, J., 1992. The 1990 conterminous U. S. AVHRR data set. *Photogrammetric Engineering and Remote Sensing* 58, 809-813.
13. Goff, S.A., Vaughn, M., McKay, S., Lyons, E., Stapleton, A.E., Gessler, D., Matasci, N., Wang, L., Hanlon, M., Lenards, A., 2011. The iPlant collaborative: cyberinfrastructure for plant biology. *Frontiers in plant science* 2.
14. Goodchild, M.F., Yuan, M., Cova, T.J., 2007. Towards a general theory of geographic representation in GIS. *International Journal of Geographical Information Science* 21, 239-260.
15. Gorokhovich, Y., Voustianiouk, A., 2006. Accuracy assessment of the processed SRTM-based elevation data by CGIAR using field data from USA and Thailand and its relation to the terrain characteristics. *Remote Sensing of Environment* 104, 409-415.
16. Hey, T., Tansley, S., Tolle, K., 2009. The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research.

17. Iacovella, S., 2012. The past, present and future of GIS: PostGIS 2.0 is here! *Linux Journal* 2012, 4.
18. Jabeur, N., McCarthy, J.D., Xing, X.T., Graniero, P.A., 2009. A knowledge-oriented meta-framework for integrating sensor network infrastructures. *Computers & Geosciences* 35, 809-819.
19. Joussaume, S., 2009. High-performance computing for climate modelling, DEISA-PRACE Symposium, Amsterdam.
20. Kardinal Jusuf, S., Wong, N.H., Hagen, E., Anggoro, R., Hong, Y., 2007. The influence of land use on the urban heat island in Singapore. *Habitat International* 31, 232-242.
21. J. Laura, W. Li, S. Rey and L. Anselin, 2015. Parallelization of regionalization heuristic in distributed computing platforms – A case study with parallel-p-compact-regions problem. *International Journal of Geographical Information Science* 29(4), 536-555.
22. Li, W., Li, L., Goodchild, M.F., Anselin, L., 2013. A geospatial cyberinfrastructure for urban economic analysis and spatial decision-making. *ISPRS International Journal of Geo-Information* 2, 413-431.
23. Li, W., Song, M, Zhou, B., CaoK., and GaoS. (2015). Performance improvement techniques for geospatial web services in a cyberinfrastructure environment. – A case study with a disaster management portal. *Computers, Environment and Urban Systems*. doi:10.1016/j.compenvurbsys.2015.04.003.
24. Li, W., Bhatia, V., & Cao, K. (2014). Intelligent polar cyberinfrastructure: enabling semantic search in geospatial metadata catalogue to support polar data discovery. *Earth Science Informatics*, 8(1), 111-123.
25. Lopes Siqueira, T. L., Ciferri, R. R., Times, V. C., & de Aguiar Ciferri, C. D. (2009, March). A spatial bitmap-based index for geographical data warehouses. In *Proceedings of the 2009 ACM symposium on Applied Computing* (pp. 1336-1342). ACM.
26. Mennis, J., 2010. Multidimensional Map Algebra: Design and Implementation of a Spatio-Temporal GIS Processing Language. *Transactions in Gis* 14, 1-21.
27. Moore, J.A., Serreze, M.C., Middleton, D., Ramamurthy, M., Yarmey, L., 2013. The Arctic Cooperative Data and Information System: Data Management Support for the NSF Arctic Research Program. *Polar Data Activities in Global Data Systems*, 19.
28. Moreno-Sanchez, R., Anderson, G., Cruz, J., Hayden, M., 2007. The potential for the use of Open Source Software and Open Specifications in creating Web-based cross-border health spatial information systems. *International Journal of Geographical Information Science* 21, 1135-1163.
29. NSF, 2009. NSF-supported research infrastructure: Enabling discovery, innovation and learning. NSF, 148 pp.
30. Pelekis, N., Theodoulidis, B., Kopanakis, I., Theodoridis, Y., 2004. Literature review of spatio-temporal database models. *The Knowledge Engineering Review* 19, 235-274.

31. Peters-Lidard, C.D., Houser, P.R., Tian, Y., Kumar, S.V., Geiger, J., Olden, S., Lighty, L., Doty, B., Dirmeyer, P., Adams, J., 2007. High-performance Earth system modeling with NASA/GSFC's Land Information System. *Innovations in Systems and Software Engineering* 3, 157-165.
32. Rinner, C., Hussain, M., 2011. Toronto's Urban Heat Island—Exploring the Relationship between Land Use and Surface Temperature. *Remote Sensing* 3, 1251-1265.
33. Ritter, N., Ruth, M., 1997. The GeoTiff data interchange standard for raster geographic images. *International Journal of Remote Sensing* 18, 1637-1647.
34. Saadat, H., Adamowski, J., Bonnell, R., Sharifi, F., Namdar, M., Ale-Ebrahim, S., 2011. Land use and land cover classification over a large area in Iran based on single date analysis of satellite imagery. *ISPRS Journal of Photogrammetry and Remote Sensing* 66, 608-619.
35. Saeed, N., Reza, D.M., 2003. Spatio-Temporal Modeling of Dynamic Phenomena in GIS, *ScanGIS*, pp. 215-225.
36. Segev, A., Shoshani, A., 1993. A Temporal Data Model Based on Time Sequences.
37. Shan, J., Zaheer, M., Hussain, E., 2003. Study on Accuracy of 1-Degree DEM Versus Topographic Complexity Using GIS Zonal Analysis. *Journal of Surveying Engineering* 129, 85-89.
38. Song, M. M., Li, Z., Zhou, B., Li, C. L., 2014. Cloud Computing Model for Big Geological Data Processing. *Applied Mechanics and Materials*, 475, 306-311.
39. Song, M., Zhou, B., Zhou, Z., 2012. The interoperation framework of ocean observation data using spatial information service. In: 2012 2nd International Conference on Computer Science and Network Technology (ICCSNT), IEEE, Changchun, China, pp. 1607-1611.
40. Wang, S., Armstrong, M.P., Ni, J., Liu, Y., 2005. GISolve: A grid-based problem solving environment for computationally intensive geographic information analysis, *Challenges of Large Applications in Distributed Environments*, 2005. CLADE 2005. Proceedings. IEEE, pp. 3-12.
41. Wang, S., 2010. A CyberGIS framework for the synthesis of cyberinfrastructure, GIS, and spatial analysis. *Annals of the Association of American Geographers* 100, 535-557.
42. Whiteaker, T.L., Maidment, D.R., Gopalan, H., Patino, C., McKinney, D.C., 2007. Raster-network regionalization for watershed data processing. *International Journal of Geographical Information Science* 21, 341-353.
43. Wu, K., Li, K., 2007. Association between esophageal cancer and drought in China by using Geographic Information System. *Environment International* 33, 603-608.
44. Yang, C., Goodchild, M., Huang, Q., Nebert, D., Raskin, R., Xu, Y., Bambacus, M., Fay, D., 2011a. Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing? *International Journal of Digital Earth* 4, 305-329.
45. Yang, C., Li, W., Xie, J., Zhou, B., 2008. Distributed geospatial information processing: sharing distributed geospatial resources to support Digital Earth. *International Journal of Digital Earth* 1, 259-278.

- 1 46. Yang, C., Wong, D.W., Yang, R., Kafatos, M., Li, Q., 2005. Performance-improving
2 techniques in web-based GIS. *International Journal of Geographical Information*
3 *Science* 19, 319-342.
- 4 47. Yang, C.W., Wu, H.Y., Huang, Q.Y., Li, Z.L., Li, J., 2011b. Using spatial principles to
5 optimize distributed computing for enabling the physical science discoveries.
6 *Proceedings of the National Academy of Sciences of the United States of America* 108,
7 5498-5503.
- 8 48. Yuan, M., 1996. Temporal GIS and spatio-temporal modeling, *Proceedings of Third*
9 *International Conference Workshop on Integrating GIS and Environment Modeling*,
10 Santa Fe, NM.
- 11 49. Yuan, M., 2001. Representing Complex Geographic Phenomena in GIS. *Cartography*
12 *and Geographic Information Science* 28, 83-96.
- 13 50. Zhang J., Simin, Y., 2013. Parallel Zonal Summations of Large-Scale Species
14 Occurrence Data on Hybrid CPU-GPU Systems. Technical report. Online at
15 http://wwwcs.ccny.cuny.edu/~jzhang/papers/zonalstat_tr.pdf.
- 16 51. Zhang, T.; Tsou, M., 2009. Developing a grid-enabled spatial Web portal for Internet
17 GIServices and geospatial cyberinfrastructure. *International Journal of Geographical*
18 *Information Science* 23, 605-630.
- 19 52. Zhao, G., Bryan, B.A., King, D., Song, X., Yu, Q., 2012. Parallelization and
20 optimization of spatial analysis for large scale environmental model data assembly.
21 *Computers and Electronics in Agriculture* 89, 94-99.
- 22 53. Zhou, Z., Zhou, B., Li, W., Griglak, B., Caiseda, C., Huang, Q., 2009. Evaluating query
23 performance on object-relational spatial databases. In: *2nd IEEE International*
24 *Conference on Computer Science and Information Technology (ICCSIT 2009)*, IEEE,
25 Beijing, China, pp. 489-492.