SCIDB BASED FRAMEWORK FOR STORAGE AND ANALYSIS OF REMOTE SENSING BIG DATA

A. Joshi[1, *], E. Pebesma[2], R. Henriques[3], M. Appel[2]

[1] Survey Department, Kathmandu, Nepal – theabhash@gmail.com
[2] Institute for Geoinformatics, University of Münster, Muenster, Germany
[3] NOVA Information Management School, Lisbon, Portugal

Commission V, WG V/7 & Commission IV, WG IV/6

KEY WORDS: Big Data, Remote Sensing, Array Database, SciDB, Parallel Processing, Time series analysis

ABSTRACT:

Earth observation data of large part of the world is available at different temporal, spectral and spatial resolution. These data can be termed as big data as they fulfil the criteria of 3 Vs of big data: Volume, Velocity and Variety. The size of image in archives are multiple petabyte size, the size is growing continuously and the data have varied resolution and usages. These big data have variety of applications including climate change study, forestry application, agricultural application and urban planning. However, these big data also possess challenge of data storage, management and high computational requirement for processing. The solution to this computational and data management requirements is database system with distributed storage and parallel computation.
In this study SciDB, an array-based database is used to store, manage and process multitemporal satellite imagery. The major aim of this study is to develop SciDB based scalable solution to store and perform time series analysis on multi-temporal satellite imagery. Total 148 scene of landsat image of 10 years period between 2006 and 2016 were stored as SciDB array. The data was then retrieved, processed and visualized. This study provides solution for storage of big RS data and also provides workflow for time series analysis of remote sensing data no matter how large is the size.

1. INTRODUCTION

Laney (2001) defined big data as data characterized by the 3Vs: Volume, Velocity, and Variety. That is, they are large in size, speed of generation of new data is rapid and have variety of structure. Based on above definition Remote Sensing data can be termed as big data. The massive amount of earth observation data is now available in the archive which has been collected by different sensors for a long time. National imagery archives are storing terabytes of data every day and total stored imagery volume will grow to the order of Exabyte (OGC, 1999). Currently, this data is increasing at an exceptionally fast rate with the advent of the new sensor with varied spectral, spatial and temporal resolutions. These remote sensing data of large part of the world is big wealth to model the earth. It can be used to monitor environmental events, monitor natural disasters and study climate change. Other application area includes forestry, urban planning, land management, food security. However, these Big Remote Sensing (RS) data also poses the significant challenge of management, processing, and interpretation (Ma, et al., 2015). Recent research trends show the development of processing techniques for these data, such as time series processing methods to detect change (Verbesselt, Zeileis, & Herold, 2012), identify land cover (Clark, Aide, Grau, & Riner, 2010). Nevertheless, there is still big challenge in managing these data and fulfil high computation requirement to process them.

Generally, these RS big data are stored in files and most scientific data analysis methods for these data are file-based. But as the volume of the data increases, there arises the problem not only of data management but also of computational resource. Scientific community demands for the development of novel way in order to manage these enormous data and support distributed computation to meet high computational requirement of those data.

Relational database management systems (RDBMS) have been successful in addressing storage and analysis requirement of the varied business world from a long time. However, RDBMS are showing limitation when there is need of horizontal scalability and distributed computation (Jacobs, 2009), which is an essential requirement for RS data. Moreover, remote sensing data has an array like structure and it is advantageous to store the data in an array structure, to perform many RS operations. Cudre-Mauroux et al. (2010) demonstrate that the array-based database outperforms mature MySQL database for analysis Astronomical data. Thus array-based database with distributed storage and distributed computation has potentialities to manage and process big RS data.

SciDB is an open source multidimensional array based database and supports distributed storage, parallel processing, sparse array storage and user defined function and data types (Stonebraker, Brown, Poliakov, & Raman, 2016). Planthaber et al. (2012) successfully tested SciDB to store and perform basic analysis on Modis Level 1 data. Appel & Pebesma (2016) have developed the workflow to store and retrieve three and four dimensional array of earth observation data in SciDB in an easy and reproducible way.

* Corresponding author

In this context, this study aims to develop SciDB based scalable solution to store multi temporal landsat image and perform time series analysis for change detection.

2. DATA USED

Image acquired by the Landsat Enhanced Thematic Mapper Plus (ETM+) sensor onboard the Landsat 7 satellite was used in this research study. This is a moderate resolution sensor built and operated by National Aeronautics and Space Administration (NASA). Landsat 7 was launched in 1999 and is continuously providing global data with 16-Day repeat cycle (USGS, 2016). Landsat 7 data collected after May 2003 have data gaps due to the failure of the Scan Line Corrector (SLC). This data is called SLC-off data. The data used in this study is SLC-off and has some missing scanned lines due to this hardware failure.

Image of the area between Nepal and India (WRS Path: 144, WRS Row: 040) was used. 148 image scene for different dates was used for the study.

Image captured between 7th July 2006 to 9th July 2016 and having cloud cover less than 80% was used for the experiment. There are 256 time series during the period.

3. METHODOLOGY

3.1 SciDB Database Setup

SciDB is currently supported only on Linux operating system. Interaction with SciDB server is done through iquery which is default SciDB client or by language binding using R or python. Two languages are available in SciDB: Array Query Language (AQL) and Array Functional Language (AFL). AQL is SQL-like query language whereas AFL is a functional language for SciDB. AQL is compiled into AFL.

SciDB itself has limited analysis capability but it can be extended using plugins that allow running script of powerful analytic language R and python inside SciDB array. R_exec (Lewis, 2016) plugin of SciDB provides a way to run R script inside SciDB arrays.

SciDB and other necessary plugins were installed using docker image of SciDB. (Appel, scidb-eo, 2016). The images contain SciDB, the scidb4geo extension for space-time arrays, a GDAL driver to upload and download Earth Observation datasets and r_exec plugin.

Scidb4gdal and Scidb4geo plugins were installed in order to facilitate conversion between time-service imagery to the multidimensional SciDB array and SciDB array to raster. Particularly Scidb4geo plugin (reference) stores spatial and temporal reference information of the time series satellite imagery to SciDB's system catalog. Scidb4gdal is a GDAL driver implements read and write access to SciDB array. R_exec plugin was used to run R scripts inside SciDB chunks. Communication to the server was done by Secure Shell(SSH) protocol.

3.2 Loading Data to SciDB and Restructuring it

Data in SciDB are stored in an n-dimensional sparse array. SciDB array is created by specifying its dimensions and attributes of the array. For example a 3-dimensional SciDB array may have x, y and z dimensions with values (0,1,2,…,20),(1,2,3,…,50) and (alfa,beta,…) respectively.

SciDB divides the data into smaller portions called chunk and each SciDB instance is responsible for storing and running queries on chunk (SciDB User's Guide, 2013). Because of this uniform distribution of storage and workload SciDB is able to deliver scalable performance on very large data sets.

The data was uploaded in SciDB using the gda_translate function of the gdalUtils library in R interface. The date of the image scene was extracted from its name and the image was placed in multidimensional array accordingly.

Only two bands from the available image were loaded in SciDB separately and they were joined later to make a single array. The AFL query to join band 3 and band 4 is:
store(join(LS3,LS4),LS)
The figure below represents storage of landsat image as SciDB array

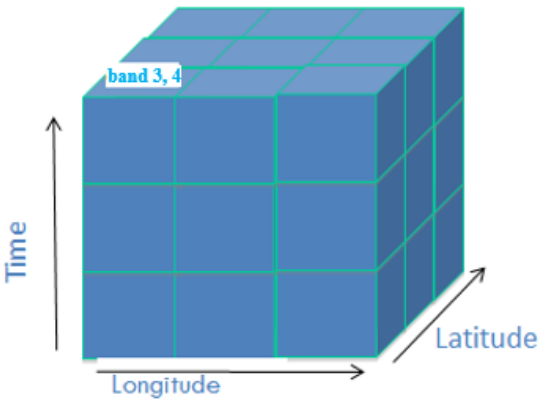

Figure 1. Image Storage as SciDB Array

After that, the repart operator in AFL was used to restructure data in chunk size of 60*60*256. It means each chunk stores complete time series of 60 rows and 60 columns. To run change monitoring using the r_exec plugin it is necessary that each chunk contains complete time series. Thus it is necessary to set t dimension as 256 to encompass all the time series in a single chunk. It is recommended to store roughly 10 to 20 MB of data in each chunk to optimize the performance of the SciDB array (SciDB User's Guide, 2013). Considering this, the value of row and columns was selected as 60.

3.3 Normalized Difference Vegetation Index (NDVI) Computation

Normalized differential vegetation index is the most frequently used index for vegetation studies. NDVI is calculated from the visible and near-infrared light reflected by vegetation. Chlorophyll pigment present in plant leaf absorbs a major portion of the visible spectrum of light for photosynthesis. However, it does not absorb NIR and some portion of it is transmitted and rest is reflected. This reflected NIR is captured in remote sensing and used for the study of vegetation

NDVI values range from +1.0 to -1.0. Very low values of NDVI (0.1 and below) correspond to barren areas of rock, sand, or snow. Moderate values represent shrub and grassland (0.2 to 0.3), while high values indicate temperate and tropical rainforests (0.6 to 0.8).

AFL was used to compute NDVI and store the file.
store(apply(between(landsat_array_repart,2150,2050,0,4250,4150,226),ndvi,(double(band1_2)double(band1))/(double(band1_2)+double(band1))),ndvi_windowSize)

3.4 Maximum NDVI Computation

Maximum NDVI is derived from the time series NDVI array. AFL was used to subset array into the desired size, compute maximum NDVI. AFL query to compute maximum NDVI array is:
store(aggregate(between(NDVI_array,2150,2050,0,4250,4150,226),max(ndvi),x,y),ndvi_max_windowSize)

3.5 Change Monitoring

Break for Additive Seasonal and Trend (BFAST) was used for detection of change from the data. BFAST allow "detection and characterization" of change in time series (Verbesselt, Hyndman, Newnham, & Culvenor, 2010). The BFAST monitoring splits time series data into history and monitoring period. From the data of historical period, it detects and models the stable history in order to detect disturbances within newly acquired data. Different models are available for modelling the stable historical behaviour. Also to determine the size of the stable history period, different methods are available.

BFAST package of R (Verbesselt, Hyndman, Newnham, & Culvenor, 2010) was used to monitor change from the time series data. Because there was too much cloud on some day, Landsat data was not available at every 16-days interval within our study period. So we have to first create regular time series objects using bfastts function in the BFAST package. This function link data with the date information and convert data of irregular date to daily time series. The start of monitoring period was chosen as 1st Jan 2012. A season-trend model with the harmonic seasonal pattern was used as a regression modeling to detect and models the stable history. Reverse-ordered CUSUM test (ROC) was used to determine the size of the stable history period. All other default parameters were used for the processing.

BFAST monitor function was run in SciDB using the r_exec plugin. The input for this operation was a SciDB array of NDVI values. R_exec works in each chunk and give the result for the chunk. For each chunk, we first split data apart. There are many options available for it in R such as the plyr package, data table package or tapply function in the basic package. We experimented with above three and found data table was fastest so used it. We then apply bfastmonitor function on the split data. Finally, we combined the output of the bfastmonitor function performed on split data together. The output of the operation is a 1-dimensional array with its row, column, breakpoint and magnitude value as attributes. We subsequently re-dimensioned the array into a two-dimensional array using row and column value.

4. RESULT

4.1 NDVI Computation

One of the primary output of the study was a three-dimensional array of NDVI value. NDVI not only detects vegetated area from non-vegetated but also can be used to derive vegetation health and other ecosystem dynamics. In this research, NDVI was also an input for subsequent experiments. SciDB automatically ignores cells in the array where values are missing and assign it as 'NA'.
The figure 7 shows part of NDVI array visualized in R. The strips in the image are the missing scan lines.

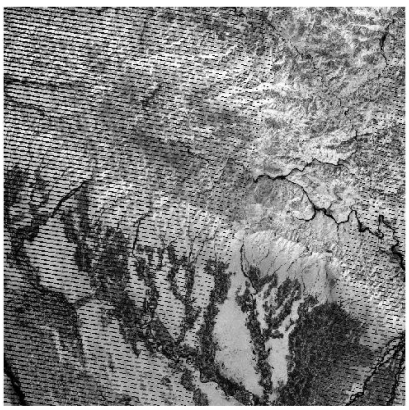

Figure 2. Subset of NDVI array visualized

4.2 Maximum NDVI Computation

Finding the maximum value of NDVI at a particular location is the simplest form of time series analysis yet very useful to summarize the time series. It is also an input for other analysis such as to compute Vegetation Condition Index (VCI). This time series analysis gives a 2-dimensional array of maximum NDVI value observed over the chosen time period. This 2-dimensional array visualized using R is presented in the figure below. In this image, there are no strips of missing scanned line as seen in NDVI image because missing lines do not overlap in all image and are removed while taking maximum value.

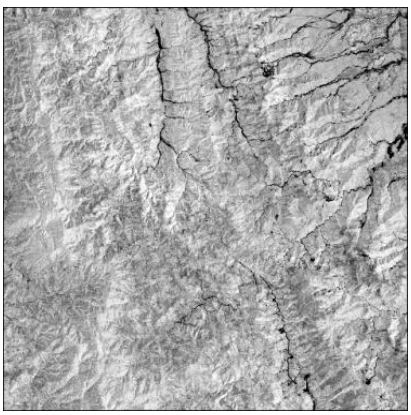

Figure 3. Maximum NDVI array visualized

4.3 Change Monitoring

Analysis to detect changes in SciDB array was performed using bfastmonitor function of bfast package. Bfastmonitor monitors change in time series by detecting disturbances in the end of time series. The output for this analysis was SciDB array with two values: the breakpoint detected with the date when this breakpoint is detected and magnitude of the median difference between the observed value and the value predicted by in the monitoring period. All the cells are assigned a value for magnitude regardless of whether the change is detected or not but no breakpoint date is assigned for the cells for which breakpoint is not detected.

The figure 4 shows the change of magnitude obtained from the function.

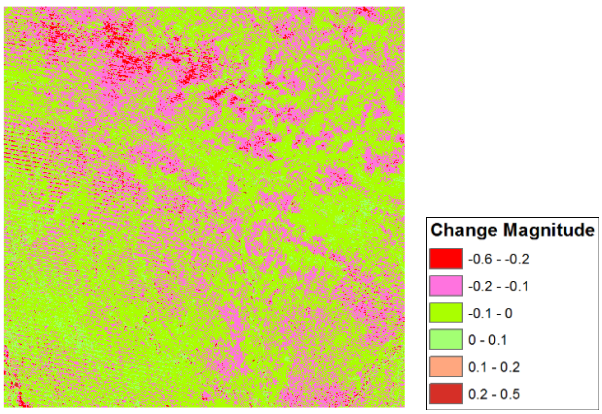

Figure 4. Change Magnitude from Bfast monitoring and legend

The red area in figure 4 represents the area where change magnitude is higher and green area of the figure suggest lower change magnitude.

Figure 5 shows the location of breakpoint detected with the year in which breakpoint is detected. It is important to note that all these changes might not be due to an actual change in the ground, which could be due to noise such as cloud in data of monitoring period. So further post processing is necessary but that is not the scope of our study.

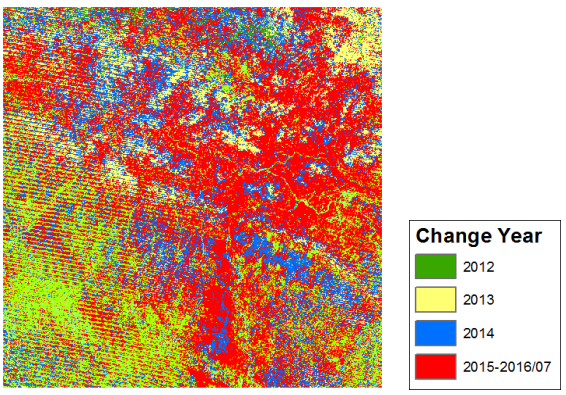

Figure 5. Breakpoint detected by Bfast Monitor with year and legend

5. CONCLUSION

System having distributed storage and horizontal scalability is the solution to ever increasing need of storage and computational requirement of remote sensing big data. In this experiment, we demonstrated a scalable solution for storage and management of multitemporal satellite imagery using SciDB. We also developed the workflow for performing for time series analysis on the image.

We also found that SciDB might not be the best solution for analysing small data as SciDB is not a mature system there is limited functionality. Also, initial system setup and data ingestion also requires time and efforts. Then again it has very good potential for management and processing of RS big data.

Further research can be conducted in the same research direction. It is necessary to investigate its applicability for running processes which requires more user interactions such as Supervised Classification. Another interesting area of study is the use of SciDB as the backend for web-based image processing system.

REFERENCES

R Core Team(b). (2016, November 7). *Package 'parallel'.* Retrieved from https://stat.ethz.ch/R-manual/R-devel/library/parallel/doc/parallel.pdf

Anyamba, A., & Eastman, J. (1996). Interannual variability of NDVI over Africa and its relation to El Niño/Southern Oscillation. *International Journal of Remote Sensing. 1;17(13),* 2533-2548.

Appel, M. (2016). *scidb-eo*. Retrieved from github: https://github.com/appelmar/scidb-eo

Appel, M., & Pebesma, E. (2016, May 11). *Scalable Earth Observation analytics with R and SciDB*. Retrieved September 2016, from r-spatial: http://r-spatial.org/r/2016/05/11/scalable-earth-observation-analytics.html

Clark, M. L., Aide, T. M., Grau, H. R., & Riner, G. (2010). A scalable approach to mapping annual land cover at 250 m using MODIS time series data: A case study in the Dry Chaco ecoregion of South America. *Remote Sensing of Environment, 114*(11), 2816-2832.

Crist, E. P., & Cicone, R. C. (1984). A Physically-Based Transformation of Thematic Mapper Data-The TM Tasseled Cap. *IEEE Transactions on Geoscience and Remote Sensing , 22(3),* 256− 263.

Cudre-Mauroux, P., Kimura, H., Lim, K.-T., Rogers, J., Madden, S., Stonebraker, M., . . . Brown, P. G. (2010). *SS-DB: A Standard Science DBMS Benchmark.* Retrieved October 2016, from http://www-conf.slac.stanford.edu/xldb10/docs/ssdb_benchmark.pdf

Dutrieux, L. (2016). *bfastSpatial*. Retrieved from github: https://github.com/loicdtx/bfastSpatial

Eddelbuettel, D. (2017). *High-Performance and Parallel Computing with R*. Retrieved from CRAN: https://cran.r-project.org/web/views/HighPerformanceComputing.html

Hausen, E. (2016). Array-database Model (SciDB) for Standardized Storing of Hyperspectral Satellite Images. Master Thesis.

Jacobs, A. (2009). The pathologies of big data. *Communications of the ACM, 52*(8), 36-44.

Kambatla, K., Kollias, G., Kumar, V., & Grama, A. (2014). Trends in big data analytics. *Journal of Parallel and Distributed Computing, 74*(7).

Karantzalos, K., Bliziotis, D., & Karmas, A. (2015). A Scalable Geospatial Web Service for Near Real-Time, High-Resolution Land Cover Mapping. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of,Volume: 8, I*, 4665 - 4674 .

Kitchin, R. (2014). Big Data, new epistemologies and paradigm shifts. *Big Data & Society, 1*(1).

Laney, D. (2001). *3D Data Management: Controlling Data Volume, Velocity, and Variety.* Retrieved from http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf

Lee, C., Gasster, S. D., Plaza, A., Chang, C.-I., & Huang, B. (2011). Recent developments in high performance computing for remote sensing: A review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 4*(3), 508-527.

Leutner, B., & Horning, N. (2017, 01 10). *RStoolbox: Tools for Remote Sensing Data Analysis*. Retrieved from CRAN: https://cran.r-project.org/web/packages/RStoolbox/index.html

Lewis, B. W. (2016). *Run R programs within SciDB queries*. Retrieved from https://github.com/Paradigm4/r_exec

Li, S., Dragicevic, S., Castro, F. A., Sester, M., Winter, S., Coltekin, A., . . . Stein, A. (2016). Geospatial big data handling theory and methods: A review and research challenges. *ISPRS Journal of Photogrammetry and Remote Sensing, 115*, 119-133.

Liu, Y., Chen, B., Yu, H., Zhao, Y., Huang, Z., & Fang, Y. (2011). Applying GPU and POSIX thread technologies in massive remote sensing image data processing. *Geoinformatics, 2011 19th International Conference on*, (pp. 1-6).

Ma, Y., Wang, L., Zomaya, A. Y., Chen, D., & Ranjan, R. (2014). Task-tree based large-scale mosaicking for massive remote sensed imageries with dynamic dag scheduling. *IEEE Transactions on Parallel and Distributed Systems, 25*(8), 2126-2137.

Ma, Y., Wu, H., Wang, L., Huang, B., Ranjan, R., Zomaya, A., & Jie, W. (2015). Remote sensing big data computing: challenges and opportunities. *Future Generation Computer Systems*, 47-60.

Mattiuzzi, M., Verbesselt, J., Hengl, T., Klisch, A., Stevens, F., Mosher, S., . . . Detsch, F. (2017, 01 10). *MODIS: Acquisition and Processing of MODIS Products*. Retrieved from CRAN: https://cran.r-project.org/web/packages/MODIS/index.html

Nickolls, J., & Dally, W. J. (2010). The GPU computing era. *IEEE micro, 30*(2).

OGC. (1999). The OpenGIS Abstract Specification-Topic 7: The Earth Imagery Case. OGC–OpenGIS Consortium.

Piatetsky, G. (2017). *Four main languages for Analytics, Data Mining, Data Science*. Retrieved from KDnuggets: http://www.kdnuggets.com/2017/02/top-stories-2017-jan.html

Planthaber, G., Stonebraker, M., & Frew, J. (2012). EarthDB: scalable analysis of MODIS data using SciDB. *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data* (pp. 11-19). ACM.

R Core Team. (2016). *The R Project for Statistical Computing*. Retrieved from R: https://www.r-project.org/about.html

SciDB User's Guide. (2013). Retrieved October 2016, from Paradigm4: http://paradigm4.com/HTMLmanual/13.3/scidb_ug/ch01s02s01.html

Stonebraker, M., Brown, P., Poliakov, A., & Raman, S. (2016). The architecture of SciDB. *International Conference on Scientific and Statistical Database Management* (pp. 1-16). Springer.

Stonebraker, M., Brown, P., Zhang, D., & Becla, J. (2013). SciDB: A database management system for applications with complex analytics. *Computing in Science & Engineering, 15*(3), 54-62.

Tan, Z., & Yue, P. (2016). A comparative analysis to the array database technology and its use in flexible VCI derivation. *016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, (pp. 1-5). Tianjin.

USGS. (2016). Retrieved January 2017, from Landsat: https://landsat.usgs.gov/

Verbesselt, J., Hyndman, R., Newnham, G., & Culvenor, D. (2010). Detecting trend and seasonal changes in satellite image time series. *Remote sensing of Environment 114, no.1*, 106-115.

Verbesselt, J., Zeileis, A., & Herold, M. (2012). Near real-time disturbance detection using satellite image time series. *Remote Sensing of Environment, 123*, 98-108.