Big Earth Observation Data Analytics: Matching Requirements to System Architectures

Gilberto Camara
INPE - National Institute for Space Research
Av dos Astronautas 1758
Sao Jose dos Campos, Brazil
gilberto.camara@inpe.br

Luiz Fernando Assis
National Institute for Space Research
Av dos Astronautas 1758
Sao Jose dos Campos, Brazil
luizffga@dpi.inpe.br

Gilberto Ribeiro
National Institute for Space Research
Av dos Astronautas 1758
Sao Jose dos Campos, Brazil
gilberto.ribeiro@inpe.br

Karine Reis Ferreira
National Institute for Space Research
Av dos Astronautas 1758
Sao Jose dos Campos, Brazil
karine.ferreira@inpe.br

Eduardo Llapa
National Institute for Space Research
Av dos Astronautas 1758
Sao Jose dos Campos, Brazil
eduardo.llapa@inpe.br

Lubia Vinhas
National Institute for Space Research
Av dos Astronautas 1758
Sao Jose dos Campos, Brazil
lubia.vinhas@inpe.br

ABSTRACT

Earth observation satellites produce petabytes of geospatial data. To manage large data sets, researchers need stable and efficient solutions that support their analytical tasks. Since the technology for big data handling is evolving rapidly, researchers find it hard to keep up with the new developments. To lower this burden, we argue that researchers should not have to convert their algorithms to specialised environments. Imposing a new API to researchers is counterproductive and slows down progress on big data analytics. This paper assesses the cost of research-friendliness, in a case where the researcher has developed an algorithm in the **R** language and wants to use the same code for big data analytics. We take an algorithm for remote sensing time series analysis on compare it use on map/reduce and on array database architectures. While the performance of the algorithm for big data sets is similar, organising image data for processing in Hadoop is more complicated and time-consuming than handling images in SciDB. Therefore, the combination of the array database SciDB and the **R** language offers an adequate support for researchers working on big Earth observation data analytics.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.2.8 [**Database Applications**]: [Spatial databases and GIS, Scientific databases, Image databases]

Keywords

Earth Observation, Array Databases, Big Data Analytics

BigSpatial16, October31–November 03 2016, Burlingname, CA, USA

ISBN 978-1-4503-4581-1/16/10. . . $15.00

DOI: http://dx.doi.org/10.1145/3006386.3006393

1. INTRODUCTION

Earth observation (EO) satellites produce vast amounts of geospatial data. The Landsat archive holds over five million images of the Earth's surface, with about 1 PB of data. New satellites from Europe, USA, China, Brazil, and India generate yearly as much data as one Landsat satellite in a decade. Most space agencies have adopted an open data policy, making unprecedented amounts of satellite data available for research and operational use. This data deluge has brought about a major challenge for Geoinformatics research: *How to design and build technologies that allow the EO community to analyse big data sets?*

When scientists use big EO data they face the burden of organising thousands of files, downloaded from the space agencies archives. To manage such large data sets, researchers need stable and efficient solutions that support their analytical tasks. To choose a solution is hard because the technology for large data handling and analytics is evolving. Alternatives include MapReduce-based solutions such as Google Earth Engine [10], object-relational DBMS extensions such as Rasdaman [2] and distributed multidimensional array databases such as SciDB [26]. Since each of these architectures takes on a different approach, understanding the benefits and drawbacks of each one helps researchers choose what best fits their needs.

Given the diversity of options, researchers would gain from documented experience that helps them to assess how proposed big data architectures fit the needs of geospatial data analysis. Recent papers describe algorithms required for EO analysis [27] [21] and report case studies using specific architectures [23][18]. However, to make progress on big geospatial data analysis, we need to engage the large community of remote sensing researchers. In this paper, we consider how big EO data architectures can support the needs of data analytics. Our paper examines ways to cut the effort required for researchers to develop and validate algorithms for extracting information for big EO data.

We take the viewpoint that architectures should serve applications, and not the other way around. To clarify the researcher's problem, we consider the needs for an important

EO application: land use and land cover change (LUCC) analysis. We propose an architecture based on open-source tools that combines array databases with statistical analysis. We evaluate this design to assess how it meets the needs of EO scientists and compared with other approaches that aim to meet these needs. This paper also puts forward a set of criteria to build researcher-friendly architectures for big EO data analysis.

2. BIG EO DATA ANALYTICS FOR LAND COVER CHANGE

When designing an architecture for big EO data, one needs to consider the needs of data analytics. A common view of big EO data processing assumes that algorithms work on a pixel-by-pixel basis. In this view, massively parallel solutions based on the MapReduce model would fit this job. However, behind the simple raster geometry of remote sensing images, lies a huge diversity of processing algorithms. Many EO tasks require sophisticated methods for spatial, temporal and spatiotemporal analysis. For such applications, scientists need to balance between parallel execution and design flexibility, so that complex algorithms can be executed with acceptable performance.

To consider the needs of big EO data analytics, we focus in a demanding application: land cover change analysis. Land cover change is one of the most immediate consequences of humanity's transformation the Earth's ecosystems and landscapes. To understand the impact and extent of global land cover change, we need data from EO satellites, the only source that provides a continuous and consistent set of information about the Earth.

Current global and large-scale land cover products need to be improved. Global land cover data sets such as MODIS Land Cover, GLC2000 and GlobCover have large mismatches on the spatial distribution of their land classes [20] [15]. Land use practices are becoming subtler than a transition from one cover (e.g, forest) to another (e.g., pasture). We need to capture changes associated with forest degradation and temporary or mixed agricultural regimes [3]. Therefore, developing new analytics for land cover change analysis using big EO data is as important as having efficient data management methods.

To better understand the requirements of big EO data analytics, consider a conceptual view of the problem. Earth observation satellites revisit the same place at regular intervals. Thus measures need to be calibrated so that observations of the same place in different times are comparable. After adjustment, the observations are organised in regular intervals; each measure from an imaging sensor maps to a point in a three-dimensional array in space-time (Figure 1). Let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of remote sensing images which shows the same region at n consecutive times $T = \{t_1, t_2, \ldots, t_n\}$. Each location <*x, y, t*> of a pixel in an image (latitude, longitude, time) maps to a <*i, j, k*> position in a 3D array. Each array position <*i, j, k*> has to a set of attributes values $A = \{a_1, a_2, \ldots, a_m\}$ which are the sensor measurements at each location in space-time (see Figure 1). For optical sensors, these observations are proportional to Earth's reflexion of the incoming solar radiation at different wavelengths of the electromagnetic spectrum. Therefore, a 3D array is an appropriate conceptual model for big EO data.

An example of big EO data analysis is the work by Hansen et al. [13]. Using more than 650,000 LANDSAT images and processing more than 140 billion pixels, the authors compared data from 2000 to 2010 to produce maps of global forest loss. A pixel-based classification algorithm was used to process each image to detect forest cover. The results for 2000 and 2010 were compared to account for forest loss during the 2000-2010 decade. The method classifies each 2D image one by one. The authors compare the results for different time instances, using a *space-first, time-later* approach.

By contrast, methods such as the time-weighted dynamic time warping (TWDTW) [19], TIMESTAT [14] and BFAST [28] work on remote sensing time series to extract long-term information for each pixel. These algorithms work on individual time series and combine the results for selected periods to generate classified maps. We call this the *time-first, space-later* approach.

The benefits of remote sensing time series analysis arise when the temporal resolution of the big data set is able to capture the most important changes. Here, the temporal autocorrelation of the data can be stronger than the spatial autocorrelation. Given data with adequate repeatability, a pixel will be more related to its temporal neighbours than to its spatial ones. In this case, *time-first, space-later* methods lead to better results than the *space-first, time-later* approach.

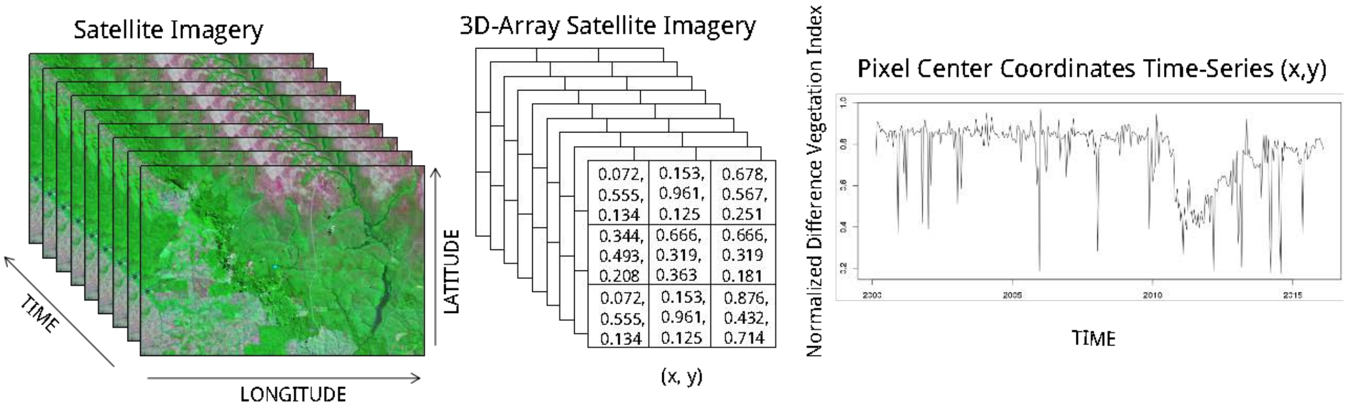
Using the 3D array metaphor, scientists can approach the classification problem using both the *space-first, time-later* and the *time-first, space-later* approaches. To enable researchers to develop innovative analytical methods for big EO data, the system architecture needs to support both approaches.

3. ARCHITECTURES FOR BIG EO DATA ANALYTICS

Progress on big EO data analytics depends on researchers developing and sharing new methods. One crucial observation is that researchers are most productive when working on familiar computing environments. Scientists like to test new ideas on small and well-known data sets. Only after they are satisfied with the experiments, they move up to work with big data. Therefore, an architecture for big Earth observation data analytics should meet important needs of the research community, described below.

1. *Analytical scaling*: provide support for the full cycle of research, allowing algorithms developed at the desktop to run on big databases with minor changes.
2. *Software reuse*: allow researchers to adapt existing methods for big data with minimal reworking.
3. *Collaborative work*: enable results to be shared with the scientific community.
4. *Replication*: encourage research teams to build their own infrastructure.

Data scientists are conservative in their choice of tools. They prefer to work on tools with a simple software kernel where they can add new packages that encapsulate new analytical methods. A prime example is the **R** suite of statistical tools [24]. **R** provides a wide variety of statistical and graphical tools, including spatial analysis, time-series


Satellite Imagery
TIME
LATITUDE
LONGITUDE
3D-Array Satellite Imagery
(x, y)
Normalized Difference Vegetation Index
Pixel Center Coordinates Time-Series (x,y)
1.0
0.8
0.6
0.4
0.2
2000
2005
2010
2015
TIME

0.072, 0.555, 0.134	0.153, 0.961, 0.125	0.678, 0.567, 0.251
0.344, 0.493, 0.208	0.666, 0.319, 0.363	0.666, 0.319 0.181
0.072, 0.555, 0.134	0.153, 0.961, 0.125	0.876, 0.432, 0.714

Figure 1: A Normalized Difference Vegetation Index (NDVI) time series.

analysis, classification, clustering, and data mining for many disciplines (e.g. hydrology, ecology, soil science, agronomy). It is extensible through user-contributed packages. Thus, **R** is the *lingua franca* of data analytics. It provides methods and tools in an open source environment and allows research reproducibility. Using **R** as their primary tool for big data analytics, researchers can thus scale up their methods, reuse previous work, and collaborate with their peers.

Given these needs, we propose an architecture adapted to researchers and not the other way around. We comply with the best practises of the scientific community by using a known programming environment such as **R**, in which scientists can execute their algorithms directly in big data servers. The two main options for researchers that want to use **R** for big EO analytics are MapReduce architectures such as Apache Hadoop or array databases such as SciDB [26].

One trend for big data analysis is the MapReduce model, whose most popular open source implementation is Apache Hadoop (http://hadoop.apache.org/). The MapReduce model has been motivated by parallel applications such as text queries and its main goal is to support task execution using a scalable cluster of computing nodes[5]. Using the Hadoop API, programmers can adapt the MapReduce model by customising how inputs and outputs are split into key/value pairs. For vector-based geospatial data, researchers have developed tools such as Hadoop-GIS and SpatialHadoop solutions. They require an extra preprocessing overhead to allow GIS functions to process data [1, 6].

The most used MapReduce-based tool for working with big Earth Observation data is Google Earth Engine (GEE) [10]. GEE offers programming interfaces that support only pixel-based image processing. Its methods neither support region-based methods such as image segmentation, nor allow large-scale time series analysis. These design decisions limit researchers who would like to perform object-based image analysis on large data sets or to perform time series data analysis. Also, the API of Google Earth Engine is proprietary, and thus researchers have to convert their existing methods to its language.

We consider array databases to be the main alternative to MapReduce-based tools. Array DBMS such as SciDB [26] rely on the mathematical concept of array, allowing interoperability at the algebraic level [4]. They reduce the impedance mismatch between the data model (raster), the storage model (arrays) and the analysis functions such as linear algebra and image processing [11]. Array databases split large volumes of data in distributed servers in a "shared nothing" way. A big array is broken into "chunks" that are distributed among different servers; each server controls its local data storage. Arrays are multidimensional and uniform, as each array cell holds the same user-defined number of attributes. Since arrays are a natural data structure to store Earth Observation images, using SciDB researchers and institutions can break the "image-as-a-snapshot" paradigm. Entire collections of image data fit into single spatiotemporal arrays.

Both the SciDB array database and the **R** statistical language are open-source and together provide computational support for parallelising complex analysis. SciDB has an **R** interface that allows researchers to run their **R** algorithms for extracting information from large remote sensing data sets (Figure 2). The SciDB design is ”shared-nothing” and scalable; it is possible to add more dedicated servers to an existing configuration. Combining array DBMS with statistical computing is a natural solution for EO applications.

In terms of effort for setting up the architecture, organising a MapReduce-base environment for big EO data needs more work than the equivalent array database solution. The main reason is that MapReduce has been designed to work with key/value pairs, requiring an extra preprocessing step of breaking up the data. In contrast, whole collections of images are mapped directly into array databases. These issues are discussed in more detail in the next section, when we compare the two approaches.

4. THE COST OF BEING RESEARCHER-FRIENDLY

In this section, we evaluate the performance cost of having an architecture focused on researcher needs. We focus on remote sensing time series analysis. We chose a time-consuming algorithm: the Time-Weighted Dynamic Time Warping (TWDTW) method for land cover mapping [19]. Besides TWDTW, algorithms for analysis of remote sensing time series include time series reconstruction [25], detecting trend and seasonal changes [28], extracting seasonality information [14], land cover mapping [12], detecting forest disturbance and recovery [16], crop classification [22] and crop expansion and intensification [9]. These innovative methods

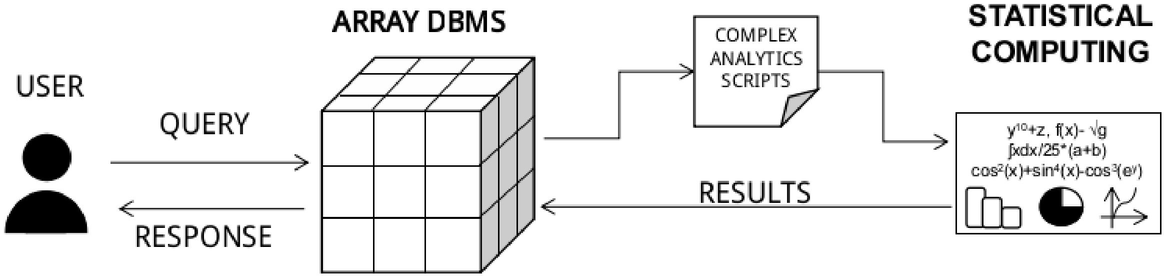


Figure 2: Proposed architecture for big EO data analytics

show that extracting information from remote sensing time series is one of the most promising research trends in big Earth observation data analysis.

Research on time series data mining shows that methods based on dynamic time warping (DTW) have achieved good results [7]. The algorithm compares two time series and finds their optimal alignment, providing a dissimilarity measure. DTW provides a robust distance measure for comparing time series [17]. The DTW algorithm works well for shape matching, but is not suited *per se* for remote sensing time series classification. Each land cover class has a distinct phenological cycle which is relevant for space-time classification [29]. A good time-series land cover classifier needs to balance between shape matching and temporal alignment. To avoid mismatches, in [19] we introduce a time constraint in TWDTW to distinguish between different land use and land cover classes. This method is flexible to account for multiyear crops, single cropping and double cropping. It is also robust to account for other land cover types such as forest and pasture and works with a small amount of training samples.

By taking a complex method such as TWDTW as the basic algorithm for our comparison, we develop a realistic case study. We take the case where a researcher has developed a new method, validates it in small data sets and wants to use it for exploring big data. Our task is to evaluate the cost of using a code developed in **R** and apply it to big data with minor adjustments. We compare two possible approaches: using **R** together with SciDB and with Hadoop. The evaluation considers both the execution time and also the costs of building and adapting each environment.

4.1 Experimental Setup and Datasets

The evaluation uses the MOD13Q1 product from NASA's MODIS collection 5 [8]. This data set has 18,000 images covering Brazil from 2000 to 2016 compiled every 16 days. Each pixel has a 250 m x 250 m spatial resolution. The total data size is about 10 TB. Our cluster has five servers using 24 CPUs of 2.40GHz, 16 disks of 1.1TB, 94 GB RAM of 2GHz AMD Opteron(tm) Processor 4171 HE and 3.4 GB RAM memory. They run Ubuntu 12.04.5 LTS (64 bit) and a switch of 1Gb interfaces all of the servers. This is a typical infrastructure for a medium-size research laboratory to work with big Earth observation data.

We built two experimental setups: one with SciDB and other with Hadoop. For SciDB, all 18,000 MOD13Q1 satellite images covering Brazil from 2000 to 2016 were loaded as 2 dimensional arrays of pixels in a snapshot mode in SciDB; they were then stacked into a 3D array. SciDB splits this data set into subsets which are distributed to the data servers. Each server runs several instances of SciDB. The SciDB coordinator instance organises the query execution and is also responsible for client-server communications. The remaining instances take part in the distributed processing of data queries. This organisation take advantage of the distributed CPUs, memory and disks to maximise parallel performance.

For Hadoop, we used its Streaming API which implements the MapReduce model, breaking an arbitrary set of parallel and intensive tasks into parts. The main challenge of using Hadoop's streaming API is to adapt the input and output data sets to sets of key/value pairs. Binary data sets such as remote sensing images have to be converted to flat files with binary key/value pairs. To adapt our data set to run in Hadoop, we preprocessed the MOD13Q1 satellite images, transforming them to a sequence of files containing one (location, time series) pair for each line. We ran the TWTDW algorithm to process each time series.

This strategy is a simplification of the *map/reduce* model. The mapping is done in the preprocessing phase by breaking a 3D array into a set of time series. We had to split these time series between the different nodes. Unlike SciDB, Hadoop does not automatically split a large data set into multiple nodes. This setup does not require a *reduce* step. This adaptation of a 3D array to run in Hadoop leads to a loss of generality. In an array database such as SciDB, the algorithm can address arbitrary partitions of space-time. To run in Hadoop each type of algorithm has to perform its own preprocessing step. Therefore, in terms of researcher-friendliness, array databases are easier to work with than map/reduce environments.

For interfacing SciDB and R, we also use a streaming solution, using a SciDB operator that streams the data sets for use in **R** programs. We envisage a large "shared-nothing" cluster where each local machine has access to its own disk. The first step is to send the **R** programs to be executed by the local machines. At each node, the SciDB streaming operator reads the local data sets and processes them using a multi-core version of **R**. Data analysis takes place in a parallel version at each node of the cluster. This interface is researcher-friendly. It allow the analysis of big data sets by reusing existing algorithms.

4.2 Comparing Hadoop with SciDB for big EO data analytics

We compared the performance of the **R** version of TWTDW in SciDB and in Hadoop. In both cases, the data was broken into 64MB chunks in space-time. Since TWDTW is used for time series analysis, we fixed the time duration (380 intervals of 16 days each). We measured the performance on different 2D slices, varying from a block of 1000x1000 to one of 4000x4000 pixels. We organised the SciDB array database to optimise performance of time series analysis, favouring *time-first, space-later* methods over *space-first, time-later* algorithms. We did this to match the preprocessing of the Hadoop data sets, which also favours time series analysis.

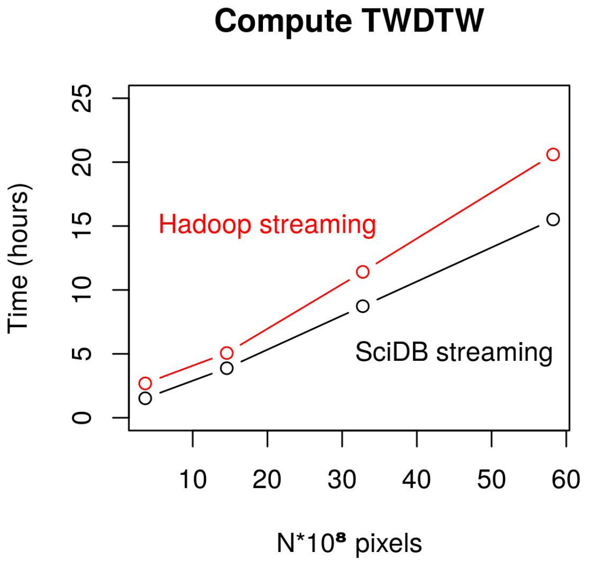


Figure 3: Comparing Hadoop and SciDB performance using the TWDTW .

Results (shown in Figure 3) show a quasi-linear scalability in both Hadoop and SciDB architectures, while performance time is similar. TWDTW is a recursive algorithm that requires a lot of in-memory processing; the method is more compute-bound than I/O bound. Since TWDTW processes each time series separately, processing time increases linearly with the number of time series. Other algorithms with different breakdowns in compute-bound and I/O bound parts will get different performances in both Hadoop and SciDB. However, we are confident our result holds in general for a large set of remote sensing time series analysis methods.

We recognise benchmarking of big data analytics remains a contentious issue. There are multiple alternatives for fine-tuning complex environments such as Hadoop. Other researchers could organise Hadoop or SciDB architectures to improve their performance measures. However, most of our performance limitations are due to our conscious choice of using **R** code instead of native APIs of Hadoop and SciDB. It is unlikely that orders-of-magnitude gains can be achieved by fine-tuning either architecture.

5. CONCLUSIONS

This paper addressed the problem of designing an architecture for big Earth observation data analytics that meets the most common needs of researchers. We argue that researchers should not have to convert their algorithms to specialised processing environments. Imposing a new API to researchers would be counterproductive and would slow down progress on big data analytics. For this reason, our main concern was to assess the cost of research-friendliness. Our scenario is a case study where the researcher has code that has already been tested in the **R** language and wants to use the same code for big data analytics.

Our restriction of using algorithms in **R** narrows the current alternatives for big EO architectures to two main options: array databases such as SciDB and map/reduce environments such as Hadoop. Both solutions provide a streaming API that can be used to run **R** code with minimal change. However, more work is required to adapt Hadoop to run **R** for EO data analysis than in the case of SciDB. The map/reduce model used by Hadoop requires preprocessing the image data into text files (or sequence files), while the array database represents image data directly. Thus, organising image data for processing in Hadoop is more complicated and time-consuming than handling images in SciDB. In terms of performance, we obtain a similar result when running the TWDTW method in both Hadoop and SciDB solutions.

When considering both the cost of data organisation and the execution time, we conclude that array databases are superior to map/reduce solutions for big EO data analytics. The combination of **R** algorithms with SciDB can be an acceptable solution to the problem of providing a friendly environment for big EO analytics. Array DBMS provide consistent data management of big EO data, while statistical computing tools support complex analytics methods.

6. ACKNOWLEDGEMENTS

The authors are supported by the São Paulo Research Foundation (FAPESP) e-science program (grant 2014-08398-6). Gilberto Camara is also supported by CNPq (grant 312151-2014-4). Eduardo Llapa is also supported by a BNDES Fundo Amazonia grant.

The authors thank Edzer Pebesma from the University of Münster (Germany), Jan Verbesselt from University of Wageningen (Netherlands) for many helpful discussions and ideas on the subject of the paper.

7. ADDITIONAL AUTHORS

Victor Maus (INPE, email: `victor.maus@inpe.br`), Alber Sanchez (INPE, email: `alber.ipia@inpe.br` and Ricardo Cartaxo Souza (INPE, email: `cartaxo@dpi.inpe.br`).

8. REFERENCES

[1] A. Aji, F. Wang, H. Vo, R. Lee, Q. Liu, X. Zhang, and J. Saltz. Hadoop GIS: A high performance spatial data warehousing system over mapreduce. *Proc. VLDB Endow.*, 6(11):1009–1020, 2013.

[2] P. Baumann, A. Dehmel, P. Furtado, R. Ritsch, and N. Widmann. The multidimensional database system RasDaMan. *ACM SIGMOD Record*, 27(2):575–577, 1998.

[3] M. Broich, M. C. Hansen, P. Potapov, B. Adusei, E. Lindquist, and S. V. Stehman. Time-series analysis of multi-resolution optical imagery for quantifying forest cover loss in Sumatra and Kalimantan, Indonesia. *International Journal of Applied Earth Observation and Geoinformation*, 13(2):277–291, 2011.

[4] P. G. Brown. Overview of SciDB: Large scale array storage, processing and analysis. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, pages 963–968, New York, NY, USA, 2010. ACM.

[5] J. Dean and S. Ghemawat. MapReduce: a flexible data processing tool. *Communications of the ACM*, 53(1):72–77, 2010.

[6] A. Eldawy and M. F. Mokbel. SpatialHadoop: A MapReduce framework for spatial data. In *IEEE 31st International Conference on Data Engineering (ICDE 2015)*, volume 1, pages 1352–1363, 2015.

[7] P. Esling and C. Agon. Time-series data mining. *ACM Computing Surveys*, 45(1):12:1–12:34, 2012.

[8] M. A. Friedl, D. Sulla-Menashe, B. Tan, A. Schneider, N. Ramankutty, A. Sibley, and X. Huang. MODIS collection 5 global land cover: Algorithm refinements and characterization of new datasets. *Remote Sensing of Environment*, 114(1):168 – 182, 2010.

[9] G. L. Galford, J. F. Mustard, J. Melillo, A. Gendrin, C. C. Cerri, and C. E. Cerri. Wavelet analysis of MODIS time series to detect expansion and intensification of row-crop agriculture in brazil. *Remote Sensing of Environment*, 112(2):576–587, 2008.

[10] N. Gorelick. Google Earth Engine. In *AGU Fall Meeting Abstracts*, volume 1, page 04, 2012.

[11] J. Gray, D. T. Liu, M. Nieto-Santisteban, A. Szalay, D. J. DeWitt, and G. Heber. Scientific data management in the coming decade. *SIGMOD Rec.*, 34(4):34–41, 2005.

[12] P. Griffiths, S. van der Linden, T. Kuemmerle, and P. Hostert. A pixel-based Landsat compositing algorithm for large area land cover mapping. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6(5):2088–2101, 2013.

[13] M. Hansen, P. Potapov, R. Moore, M. Hancher, S. Turubanova, A. Tyukavina, D. Thau, S. Stehman, S. Goetz, T. Loveland, A. Kommareddy, A. Egorov, L. Chini, C. Justice, and J. Townshend. High-resolution global maps of 21st-century forest cover change. *Science (New York, N.Y.)*, 342(2013):850–3, 2013.

[14] P. Jönsson and L. Eklundh. Timesat–a program for analyzing time-series of satellite sensor data. *Computers & Geosciences*, 30(8):833 – 845, 2004.

[15] A. Kaptué Tchuenté, J. Roujean, and S. De Jong. Comparison and relative quality assessment of the GLC2000, GLOBCOVER, MODIS and ECOCLIMAP land cover data sets at the african continental scale. *International Journal of Applied Earth Observation and Geoinformation*, 13(2):207–219, 2011.

[16] R. E. Kennedy, Z. Yang, and W. B. Cohen. Detecting trends in forest disturbance and recovery using yearly Landsat time series. *Remote Sensing of Environment*, 114(12):2897–2910, 2010.

[17] E. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge Information Systems*, 7(3):358–386, 2005.

[18] L. Krcal and S. Ho. A SciDB-based framework for efficient satellite data storage and query based on dynamic atmospheric event trajectory. In *4th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*, Seattle, WA, USA, 2015.

[19] V. Maus, G. Câmara, R. Cartaxo, A. Sanchez, F. M. Ramos, and G. R. de Queiroz. A time-weighted dynamic time warping method for land-use and land-cover mapping. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(8):3729 – 3739, 2016.

[20] I. McCallum, M. Obersteiner, S. Nilsson, and A. Shvidenko. A spatial comparison of four satellite derived 1km global land cover datasets. *International Journal of Applied Earth Observation and Geoinformation*, 8(4):246–255, 2006.

[21] S. Nativi, P. Mazzetti, M. Santoro, F. Papeschi, M. Craglia, and O. Ochiai. Big data challenges in building the Global Earth Observation System of Systems. *Environmental Modelling & Software*, 68:1–26, 2015.

[22] F. Petitjean, J. Inglada, and P. Gancarski. Satellite image time series analysis under time warping. *IEEE Transactions on Geoscience and Remote Sensing*, 50(8):3081–3095, 2012.

[23] G. Planthaber, M. Stonebraker, and J. Frew. EarthDB: scalable analysis of MODIS data using SciDB. *Proceedings of the 1st ACM SIGSPATIAL Workshop on Analytics for Big Geospatial Data - BigSpatial '12*, pages 11–19, 2012.

[24] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.

[25] G. J. Roerink, M. Menenti, and W. Verhoef. Reconstructing cloudfree NDVI composites using Fourier analysis of time series. *International Journal of Remote Sensing*, 21(9):1911–1917, 2000.

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

[27] R. R. Vatsavai, A. Ganguly, V. Chandola, A. Stefanidis, S. Klasky, and S. Shekhar. Spatiotemporal data mining in the era of big spatial data: Algorithms and applications. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*, pages 1–10. ACM, 2012.

[28] J. Verbesselt, R. Hyndman, G. Newnham, and D. Culvenor. Detecting trend and seasonal changes in satellite image time series. *Remote Sensing of Environment*, 114(1):106–115, 2010.

[29] X. Zhang, M. A. Friedl, C. B. Schaaf, A. H. Strahler, J. C. Hodges, F. Gao, B. C. Reed, and A. Huete. Monitoring vegetation phenology using modis. *Remote Sensing of Environment*, 84(3):471 – 475, 2003.