Cloud Computing in Remote Sensing: Big Data Remote Sensing Knowledge Discovery and Information Analysis

Yassine SABRI[1], Fadoua Bahja[2]
Laboratory of Innovation in Management and Engineering for Enterprise (LIMIE), ISGA Rabat, 27 Avenue Oqba, Agdal, Rabat, Morocco

Aouad Siham[3]
Mohammed V University of Rabat
Smart Systems Laboratory (SSL)
ENSIAS, Morocco

Aberrahim Maizate[4]
RITM- ESTC/CED -ENSEM, University Hassan II
Km7, El jadida Street, B.P. 8012, Oasis, Casablanca 8118

***Abstract*—With the rapid development of remote sensing technology, our ability to obtain remote sensing data has been improved to an unprecedented level. We have entered an era of big data. Remote sensing data clear showing the characteristics of Big Data such as hyper spectral, high spatial resolution, and high time resolution, thus, resulting in a significant increase in the volume, variety, velocity and veracity of data. This paper proposes a feature supporting, salable, and efficient data cube for time-series analysis application, and used the spatial feature data and remote sensing data for comparative study of the water cover and vegetation change.The spatial-feature remote sensing data cube (SRSDC) is described in this paper. It is a data cube whose goal is to provide a spatial-feature-supported, efficient, and scalable multidimensional data analysis system to handle large-scale RS data. It provides a high-level architectural overview of the SRSDC.The SRSDC offers spatial feature repositories for storing and managing vector feature data, as well as feature translation for converting spatial feature information to query operations.The paper describes the design and implementation of a feature data cube and distributed execution engine in the SRSDC. It uses the long time-series remote sensing production process and analysis as examples to evaluate the performance of a feature data cube and distributed execution engine. Big data has become a strategic highland in the knowledge economy as a new strategic resource for humans. The core knowledge discovery methods include supervised learning methods data analysis supervised learning, unsupervised learning methods data analysis unsupervised learning, and their combinations and variants.**

***Keywords*—Remote sensing; data integration; cloud computing; big data**

I. Introduction

In recent decades, the remarkable developments in Earth observing(EO) technology provided a significant amount of remote sensing(RS) data openly available [1]. This large observation dataset characterized the information about the earth surface in space, time, and spectral dimensions [2][3]. Apart form these dimensions, these data also contain many geographic features, such as forests, cities, lakes and so on, and these features could help researchers to locate their interested study areas rapidly. Now these multidimensional RS data with features have been widely used for global change detection research such as monitoring deforestation [4] and detecting temporal changes in floods [35]. However, the conventional geographic information system (GIS) tools are inconvenient for scientists to process the multidimensional RS data, because they lack appropriate methods to express multidimensional data models for analysis. And researchers have to do additional data organization work to conduct change detection analysis. For a more convenient analysis, they need a multidimensional data model which could support seamless analysis in space, time, spectral and feature [5].

Recently, many researchers have proposed using a multi-dimensional array model to organize the RS raster data [6][7]. Subsequently, they achieved the spatio-temporal aggregations capacity used in spatial on-line analytical processing (SOLAP) systems [8][9], as a data cube. Using this model, researchers can conveniently extract the desired data from the large dataset for analysis, and it reduces the burdens of data preparation for researchers in time-series analysis. However, in addition to extracting data with simple three-dimensional (3D) space-time coordinates, researchers occasionally need to extract data with some geographic features [10][11][12], which are often used to locate or mark the target regions of interest. For example, flood monitoring often needs to process multidimensional RS data which have the characteristics of large covered range, long time series and multi bands [13]. If we built all the analysis data as a whole data cube which has the lakes or river features, we could rapidly find the target study area we need by feature and analyse the multi bands image data to detect the flood situation with the time series. That makes researchers focus on their analysis work without being troubled by the data organization. This study aims to develop the spatial-feature remote sensing data cube(SRSDC), a data cube whose goal is to deliver a spatial feature-supported, efficient, and scalable multidimensional data analysis system to handle the large-scale RS data. The SRSDC provides spatial feature repositories to store and manage the vector feature data, and a feature translation to transform the spatial feature information to a query operation. To support large-scale data analysis, the SRSDC provides a work-scheduler architecture to process the sliced multidimensional arrays with dask [14].

The remainder of this paper is organized as follows. Section 2 describes some preliminaries and related works. Section 3 presents an architectural overview of the SRSDC. Section 4 presents the design and implementation of a feature data cube and distributed execution engine in the SRSDC. Section 5 uses the long time-series remote sensing production process

and analysis as examples to evaluate the performance of a feature data cube and distributed execution engine. Section 6 concludes this paper and describes the future work prospects.

II. Preliminaries and Related Work

A. Knowledge Discovery Categories

In the following, we discuss four broad categories of applications in geosciences where knowledge discovery methods have been widely used and have aroused impressive attention. For each application, a brief description of the problem from a knowledge discovery perspective is presented.

Detecting objects and events in geoscience data is important for a number of reasons. For example, detecting spatial and temporal patterns in climate data can help in tracking the formation and movement of objects such as cyclones, weather fronts, and atmosphere rivers, which are responsible for the transfer of precipitation, energy, and nutrients in the atmosphere and ocean. Many novel pattern mining approaches have been developed to analyze the spatial and temporal properties of objects and events, e.g., spatial coherence and temporal persistence that can work with amorphous boundaries. One such approach has been successfully used for finding spatio-temporal patterns in sea surface height data, resulting in the creation of a global catalogue of Mesoscale Ocean eddies. The use of topic models has been explored for finding extreme events from climate time series data. Given the growing success of deep learning methods in mainstream machine learning applications, it is promising to develop and apply deep learning methods for a number of problems encountered in geosciences. Recently, deep learning methods, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been used to detect geoscience objects and events, such as detecting extreme weather events from a climate model.

Knowledge discovery methods can contribute to estimating physical variables that are difficult to monitor directly, e.g., methane concentrations in air or groundwater seepage in soil, using information about other observed or simulated variables. To address the combined effects of heterogeneity and small sample size, multi-task learning frameworks have been explored, where the learning of a model at every homogeneous partition of the data is considered as a separate task, and the models are shared across similar tasks.

The sharing of learning is able to help in regularizing the models across all tasks and avoid the problem of over fitting. Focusing on the heterogeneity of climate data, online learning algorithms have been developed to combine the ensemble outputs of expert predictors and conduct robust estimates of climate variables such as temperature. To address the paucity of labeled data, novel learning frameworks such as semi-supervised learning, active learning, have huge potential to improving the state-of-the-art in estimation problems encountered in geoscience applications. Forecasting long-term trends of geoscience variables such as temperature and greenhouse gas concentrations ahead of time can help in modeling future scenarios and devising early resource planning and adaptation policies. Some of the existing approaches in knowledge discovery for time-series forecasting include exponential smoothing techniques, the auto regressive integrated moving average model and probabilistic models, such as hidden Markov models and Kalman filters. In addition, RNN-based frameworks such as long-short-term-memory (LSTM) have been used for long-term forecasting geoscience variables.

An important problem in geoscience application is to understand the relationships in geoscience data, such as periodic changes in the sea surface temperature over the eastern Pacific Ocean and their impact on several terrestrial events such as floods, droughts and forest fires. One of the first knowledge discovery methods in discovering relationships from climate data is a seminal work, where graph-based representations of global climate data were constructed. In the work, each node represents a location on the Earth and an edge represents the similarity between the eliminated time series observed at a pair of locations. The high-order relationships could been discovered from the climate graphs. Another kind of method for mining relationships in climate science is based on representing climate graphs as complex networks, including approaches for examining the structure of the climate system, studying hurricane activity. Recently, some works have developed novel approaches to directly discover the relationships as well as integrating objects in geoscience data. For example, one work has been implemented to discover previously unknown climate phenomena. For causality analysis, the most common tool in the geosciences is bivariate Grange analysis, followed by multi-variate Granger analysis using vector auto regression (VAR) models.

B. Knowledge Discovery Methods

As a new strategic resource for human beings, big data has become a strategic highland in the era of knowledge economy. It is a typical representative of the data-intensive scientific paradigm following experience, theory and computational models, since this new paradigm mainly depends on data correlation to discover knowledge, rather than traditional causality. It is bringing about changes in scientific methodology, and will become a new engine for scientific discovery.

Knowledge discovery of remote sensing big data lies at the intersection of earth science, computer science, and statistics, and is a very important part of artificial intelligence and data science. Its aims at dealing with the problem of finding a predictive function or valuable data structure entirely based on data and will not be bothered by the various data types and, is suitable for comprehensively analyzing the Earth's big data.

The core knowledge discovery methods include supervised learning methods, unsupervised learning methods, and their combinations and variants. The most widely used supervised learning methods use the training data taking the form of a collection of (x, y) pairs, and aims to produce a prediction y' in response to a new input x' by a learned mapping f(x), which produces an output y for each input x (or a probability distribution over y given x). There are different supervised learning methods based on different mapping functions, such as decision forests, logistic regression, support vector machines, neural networks, kernel machines, and Bayesian classifiers. In recent years, deep networks have received extensive attention in supervised learning. Deep networks are composed of multiple processing layers to learn representations of data with

multiple levels of abstraction, and discover intricate structures of the big earth data by learning its internal parameters to compute the representation in each layer. Deep networks have brought about breakthroughs in processing satellite image data, forecasting long-term trends of geoscience.

Unlike supervised learning methods, unsupervised learning involves the analysis of unlabeled data under assumptions about structural properties of the data. For example, the dimension reduction methods make some specific assumptions that the earth data lie on a low-dimensional manifold and aim to identify that manifold explicitly from data, such as principal components analysis, manifold learning, and auto encoders. Clustering is another very typical unsupervised learning algorithm, which aims to find a partition of the observed data, and mine the inherent aggregation and regularity of data. In recent years, much current research involves blends across supervised learning methods and unsupervised learning. Semi supervised learning is a very typical one, which makes use of unlabeled data to augment labeled data in a supervised learning context considering the difficulty of obtaining some geoscience supervision data. Overall, knowledge discovery of the big earth data needs to leverage the development of artificial intelligence, machine learning, statistical theory, and data science.

C. Related Work

With the growing numbers of archived RS images for Earth observation, an increasing number of scientists are interested in the spatiotemporal analysis of RS data. Many researchers proposed combining online analytical processing (OLAP) [15] technology with the GIS [16] to build a data cube. They built the multidimensional database paradigm to manage several dimension tables, periodically extracting the dimension information from the data in GIS, and achieved the ability to explore spatiotemporal data using the OLAP spacetime dimension aggregation operation. Sonia Rivest et al. deployed a spatial data warehouse based on GIS and spatial database to acquire, visualize, and analyze the multidimensional RS data [17]. Matthew Scotch et al. developed the SOVAT tool [18], using OLAP and GIS to perform the public health theme analysis with the data composed of spatiotemporal dimensions. These tools can facilitate researchers extracting data with spatiotemporal dimensions; however, their multidimensional data model is unsuitable for complicated scientific computing. Further, they did not adopt an appropriate architecture for large data processing [19][20]. Therefore, their ability to handle large-scale data is limited.

Owing to natural raster data structure of Earth observation images, the time-series imagery set can be easily transformed to multidimensional array. For example, a 3D array can represent the data with spatiotemporal dimensions. This data type is suitable for parallel processes, because a large array can be easily partitioned into several chunks for distributed storing and processing. In addition, the multidimensional array model enables a spatiotemporal auto-correlated data analysis; therefore, researchers need not be concerned about the organization of discrete image files. Thus, much research is focused on developing new analysis tools to process the large RS data based on the multidimensional array model; e.g., Gamara et al.[21] tested the performance of spatiotemporal analysis algorithms on array database architectures - SCIDB [22], which described the efficiency of spatiotemporal analysis based on the multidimensional array model, Assis et al.[23] built a parallel RS data analysis system based on the MapReduce framework of Hadoop [24], describing the 3D array with key/value pairs. Although these tools have significantly improved the computation performance of RS data analysis, they also contain some deficiencies. First, many of them focused only on analyzing the RS raster image data located by geographic coordinates, and did not provide the support of spatial feature, thereby limiting their ability to use these geographic objects in the analysis application. Next, some of these tools require analysers to fit their algorithms into specialised environments, such as Hadoop MapReduce framework [25]. This will be user unfriendly to researchers who only desire to focus on their analysis application.

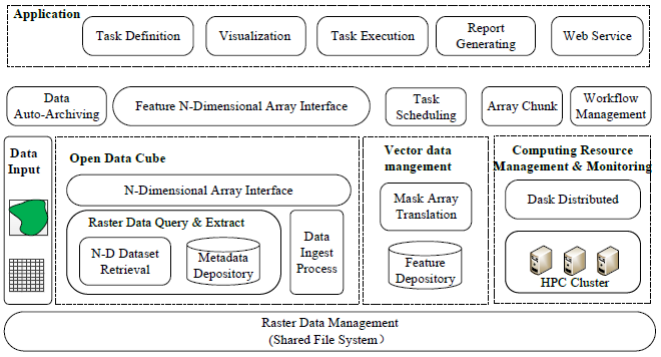

Fig. 1. The Architecture of the SRSDC.

III. Architecture Overview

A. Target Data and Environment

The SRSDC system is designed for providing the services of large RS data time-series analysis with spatial features, and it aims to manage and process the large-scale spatial feature data and satellite images seamlessly. Based on the open data cube (ODC) [26], which is a popular data cube system used for spatial raster data management and analysis, we archived large amounts of satellite data within China. These data came from different satellites including Landsat, MODIS, GaoFen(GF) and HuanJing(HJ). In addition, the SRSDC also contains many features data within China, such as lakes, forests and cities. These spatial vector data were downloaded from the official web site of OpenStreetMap [27]. Before obtaining these satellite images in the SRSDC, the geometric correction and radiometric correction for these images must be ensured. This can ensure the comparability between the images in different time, space and measurements; subsequently, the global subdivision grid can be used to partition the data into many tiles(grid files). These tiles were stored as the NetCDF format [28], which supports many analysis libraries and scientific toolkits.

B. FRSDC Architecture Overview

The SRSDC system adopted the relational database and file system to manage the spatial data. It is designed to be

scalable and efficient and provide feature support for time-series analysis. Compared with the ODC system [29], which only supports spatial raster data management and analysis as a data cube, the SRSDC supports the extraction of target satellite data as a multidimensional array with the geographic object. It could perform the spatial query operation with geographic objects, instead of locating data with only geographic coordinates. Therefore, the dataset built for analysis has geographic meaning. Thus, if researchers desire to obtain the target dataset, they only need to query data with the geographic meaning of the analysis themes, without knowing specific geographic coordinates. As shown in Fig. 1, the system is primarily composed of the data management and distributed execution engine (DEE). Data management consists of two parts, raster data management and vector data management. For the raster data, the SRSDC will archive it into a shared network file system and extract its metadata information automatically; these metadata will be stored into the metadata depository managed by ODC. For the vector data, the SRSDC stores them as geographic objects in the spatial database. After the data management, an N-Dimensional array interface is responsible for transforming the raster data and vector data to an N-Dimensional array that has the spatial feature information. Xarray [28] is used for array handing and computing. DEE is responsible for providing the computing environment and resources on high performance computing(HPC) clusters. The SRSDC use dask [24], which is a parallel computation library with blocked algorithms, for the task scheduling, distributed computing, and resource monitoring. It could help researchers to execute the analysis tasks in parallel.

IV. Design and Implementation

A. Feature Data Cube

1) Spatial feature object in FRSDC: Spatial feature is a geographic object that has special geographic meaning. It is often important for RS application, because researchers occasionally need to process the RS image dataset with geographic objects, such as the classification of an RS image with spatial features [30][31][32]. However, many RS data cube systems only provide the multidimensional dataset without features. Hence, researchers are required to perform additional work to prepare the data for analysis. For example, the ODC system [24] and the data cube based on SCIDB [22] could only query and locate the study area by simple geographic coordinates, so researchers must transform their interested spatial features to coordinate ranges one by one if they want to prepare the analysis ready data. To solve this problem, the SRSDC combined the basic N-Dimensional array with geographic objects to provide the feature N-Dimensional array for researchers, and researchers could easily organise the analysis ready N-Dimensional dataset by their interested features. Within the SRSDC, now we primarily archive the forest and lake features of China, and store them as geographic objects in a PostGIS database [33][34]. The unified modeling language(UML) class diagram in the SRSDC is shown in Fig. 2, and the description of these classes is as follows:

1) The feature class is provided for users to define their spatial feature of interest with a geographic object. It contains the feature type and the geographic object.
2) The feature type class represents the type of geographic object, such as lakes, forests, cities and so on. It contains the description of the feature type and analysis algorithm names that are suitable for the feature type.
3) The geographic object class describes the concrete vector data with geographic meaning, such as Poyang lake (a lake in China). It contains the vector data type to illustrate its geometry type.
4) The raster data type class is used to describe the type of satellite data. It contains the satellite platform, product type, and bands information.
5) The feature operation class is used to extract the feature data-cube dataset from the SRSDC. It contains the feature object, raster data type, and time horizon to build the target feature N-Dimensional array. It also provides some operation functions for users.

2) Data management: As mentioned above, data management consists of raster data management and vector data management. Because of satellite data's large volume and variety, the SRSDC uses the file system to store the raster data and manage the metadata by a relation database that contains NoSQL fields.

In the metadata depository, the SRSDC uses NoSQL fields to describe the metadata information instead of a full relation model. This is because the number of satellite sensors is increasing rapidly, and if the full relation model is used, the database schema must be expanded frequently to meet the new data sources. In contrast, NoSQL fields are more flexible in describing the metadata of satellite data that originate from different data sources. The NoSQL fields contain the time, coordinate, band, data URL, data type, and projection. Among these fields, some are used for data query, such as the time, band, coordinates, and data type. Some other fields are used for loading the data and building the multidimensional array in memory, such as data URL and projection. In addition, comparing the vector data volume (GB level) we download from the OpenStreetMap [26] with the raster data volume (TB level) we archived from several satellite data centers [32], we found that most spatial features that are represented by the vector data are not as large as the raster images; therefore we established a feature depository instead of file system to store and manage them as geographic objects. These objects may contain different geographic meanings, and we defined them as feature types.

The runtime implementation of feature data cube building and processing is shown in Fig. 3. First, the SRSDC receives the user's data request from the web portal and obtains the target geographic object by querying the feature depository. Next, it conducts a feature translation to transform the geographic object into a mask array and obtains the minimum bounding rectangle(MBR) of the feature. Subsequently, with the vertex coordinates of MBR and time horizon, the SRSDC searches for the required raster data's metadata to locate physical URLs of the raster data. Next, ODC's N-Dimensional array interface will load the raster data set from the file system and build a multidimensional array in memory. Subsequently, the mask array will be applied to masking the multidimensional array, and a new multidimensional array with features for analyzing and processing will be obtained. Finally, the SRSDC will

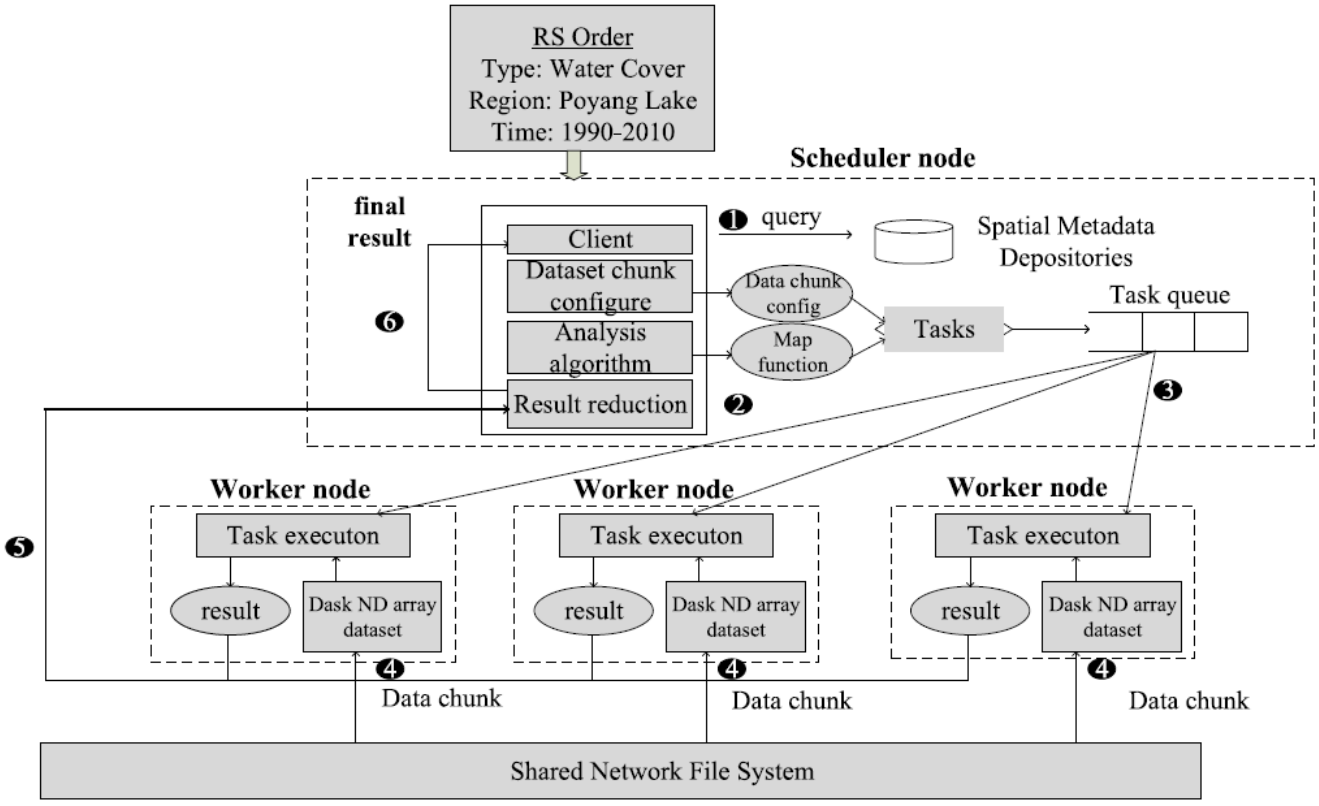

Fig. 2. Large-scale RS Analysis Processing with Distributed Executed Engine.

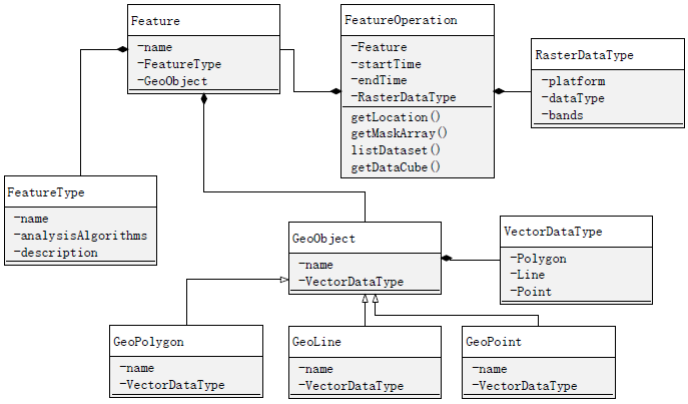

Fig. 3. The UML Class Diagram in the SRSDC.

process the data with the relevant algorithm and return the analysis results to the user.

B. Distributed Executed Engine

As an increasing number of RS applications need to process or analyze the massive volume of RS data collections, the stand-alone mode processing can not satisfy the computation requirement. To process the large-scale RS data efficiently, we built a distributed executed engine using the dask a distributed computing framework focusing on scientific data analysis. Compared with the popular distributed computing tools such as Apache Spark, dask supports the multidimensional data model natively and has a similar API with pandas and numpy. Therefore, it is more suitable for computing an N-Dimensional array. Similar to Spark, dask is also a master-slave system framework that consists of one schedule node and several work nodes. The schedule node is responsible for scheduling the tasks, while the work nodes are responsible for executing tasks. If all the tasks have being performed, these workers' computation results would be reduced to the scheduler and the final result would be obtained.

In the SRSDC, we could index the satellite image scenes by adding their metadata information to the database, and then obtain the data cube dataset (N-Dimensional arrays) from the memory for computing. However, to compute the large global dataset, we should slice the large array into the fixed-size sub-arrays called chunks for computing in the distributed environment. The SRSDC partitions these native images into seamless and massive tiles based on a latitude/longitude grid. The tile size is determined by the resolution of satellite images. For example, in the SRSDC, the Landsat data (each pixel 0.00025°) was partitioned into tiles of size 1°x 1°, and the tiles (4000x4000 pixels array) can be easily organized as a data chunk, which is suitable for the memory in the worker node. By configuring the grid number and time horizon, the chunk could be built. Further, with these data chunks, the SRSDC can transform the big dataset (N-Dimensional arrays) to several sub-arrays loaded by different worker nodes. After all the data chunks have been organized, the scheduler will assign the chunks to the workers and map the functions for

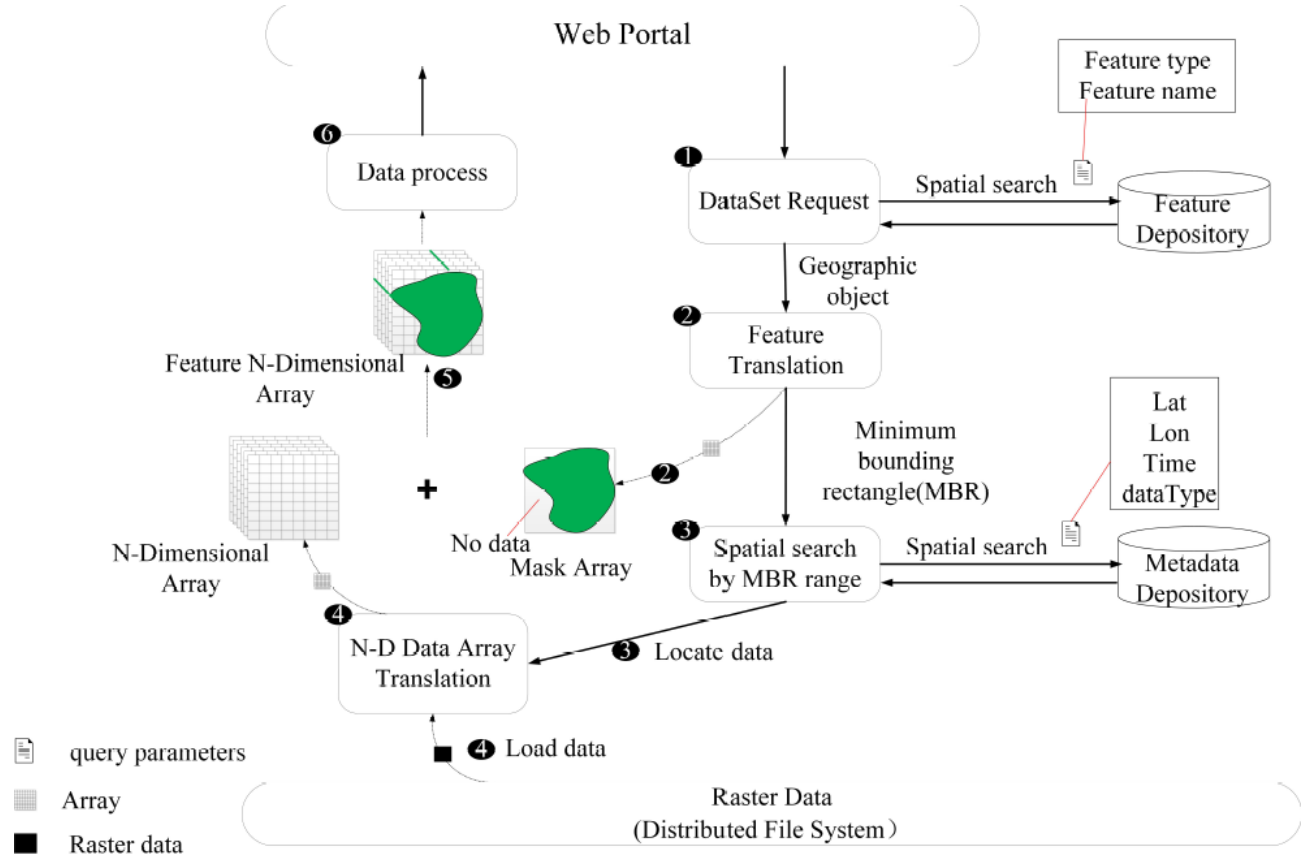

Fig. 4. Runtime Implementation of Feature Data Cube Building.

computing.

As shown in Fig. 4, the processing of large-scale time series analysis by the distributed executed engine is as follows:

1) Organize the data cube dataset by multidimensional spatial query.
2) Configure the appropriate parameters (grid number or time horizon) to organized the data chunks for workers, manage the chunks' ids with a queue.
3) Select the analysis algorithm and data chunks to compose the tasks, and assign these tasks to the worker node.
4) Check the executing state of each task in the workers; if failure occurs, recalculate the result.
5) Reduce all the results to the scheduler and return the analysis result to the client.

V. Experiments

To verify the ability of multidimensional data management and large-scale data analysis in the SRSDC, we conducted the following time-series analysis experiments focusing on spatial feature regions and compared the performance of GEE and stand-alone mode processes on the target dataset.

In this experiment, two RS application algorithms for time-series change detection have been used: NDVI for vegetation change detection and water observation from space (WOfs) for the water change detection. We built the distributed executed engine with four nodes connected by a 20 GB Infiniband network; one node for the scheduler and tree nodes for the workers. Each node was configured with Inter(R) Xeon(R)E5-2460 CPU(2.0GHz) and 32GB memory. The operating system is CentOS 6.5, and the python version is 3.6. To test the performance of the feature data cube, we selected two study regions with the special features as examples. One region for the NDVI is Mulan hunting ground, Hebei Province, China(40.7°-43.1°N, 115.8°-119.1°E), and another region for WOfs is Poyang Lake, Jiangxi Province, China. We built two feature data cube datasets for 20 years(1990-2009) with Landsat L2 data and the geographic objects. The data volume for Mulan hunting ground is approximately 138 GB and the data volume for Poyang Lake is about 96 GB. Figure 5 shows the percentage of observations detected as water for Poyang Lake over the 20-year time series. The red area represents the frequent or permanent water, and the purple area represents the infrequent water. From the result, the shape area of Poyang Lake can be observed clearly. Fig. 6 shows the annual average NDVI production on the Mulan hunting ground; Fig. 7 shows the NDVI time series result of the sampling site(41.5620°N, 117.4520°E) over 20 years. As shown, the values during 2007-2008 were abnormally below the average. This is because the average annual rainfall during this time is lower than that in normal years.

To test the processing performance of the DEE for different amounts of data, we tested time consumed by processing 6.3 GB, 12.8 GB, 49.6 GB, 109.8 GB, 138.6GB input data

for NDVI production. These data have been partitioned into 4000x4000 pixels tiles mentioned above, with which we compared the performances of the stand-alone model and DEE models:

1) stand-alone model: organize the dataset as data chunks, and process these data chunks serially with a single server.
2) DEE model: organize the large dataset as data chunks, and assign

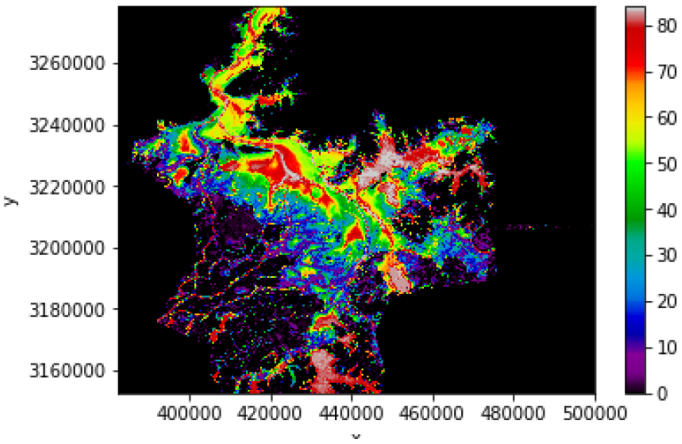

Fig. 5. Water Area of Poyang Lake over 20-year Time Series.

different workers to read these data chunks to process them in parallel with the distributed executed engine, which consists of one schedule node and three work nodes. As shown from

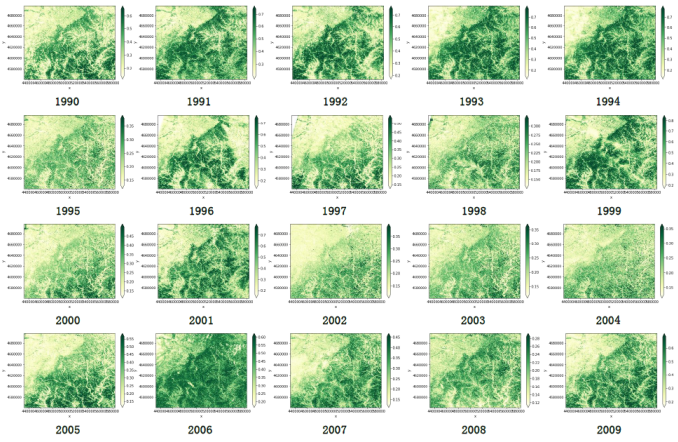

Fig. 6. The Annual Average NDVI of Mulan Hunting Ground for 20 Years.

the experimental results in Fig. 8, the DEE mode is much faster than the stand-alone mode because it can use the shared memory of clusters nodes and process the large dataset in parallel. As the process data amount increases, we also observed that the time consumed will grow nonlinearly.

This is due to the IO limit of the shared network file system and scheduling overhead. The speedup performance when generating the NDVI production with increasing numbers of work nodes also proved this point, as shown in Fig. 9. Therefore,

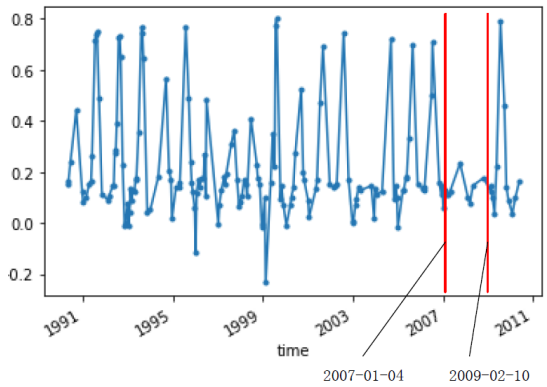

Fig. 7. A NDVI Time Series of Sampling Site on Mulan Hunting Ground.

we conclude that the SRSDC has a certain capacity to process the massive data, which is unsuitable for the memory.

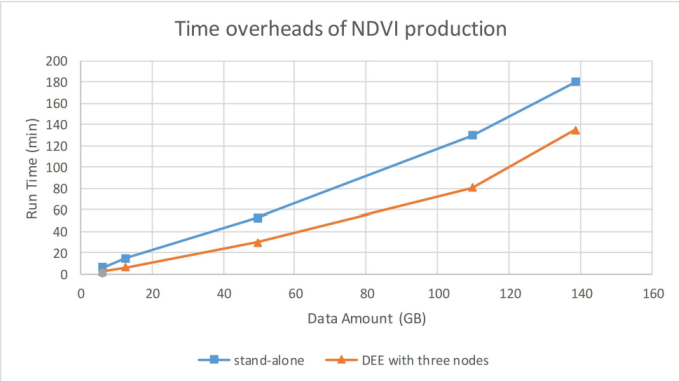

Fig. 8. Runtime of NDVI with the Increase of Data Volume.

VI. Conclusions

We have designed and tested a feature supporting, scalable, and efficient data cube for time-series analysis application, and used the spatial feature data and remote sensing data for comparative study of the water cover and vegetation change. In this system, the feature data cube building and distributed executor engine are critical in supporting large spatiotemporal RS data analysis with spatial features. The feature translation ensures that the geographic object can be combined with satellite data to build a feature data cube for analysis. Constructing a distributed executed engine based on dask ensures the efficient analysis of large-scale RS data. This work could provide a convenient and efficient multidimensional data services for many remote sensing applications [33][34]. However, it also has some limitations; for example, the image data is stored in the shared file system, and its IO performance is limited by the network.

In the future, more work will be performed to optimize the system architecture of the SRSDC, such as improving the performance of the distributed executed engine, selecting other storage methods which could ensure the process data locality, adding more remote sensing application algorithms, etc.

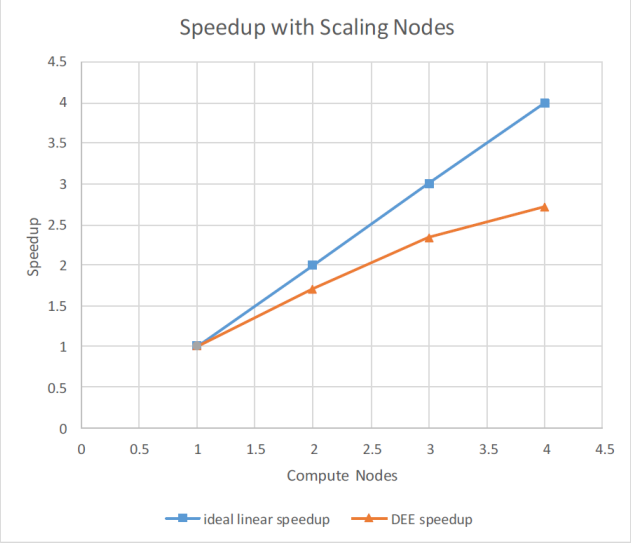

Fig. 9. Speedup for the Generation of NDVI Products with Increasing Nodes.

REFERENCES

[1] Huadong Guo, Lizhe Wang, Fang Chen, et al. Scientific big data and digital earth. Chinese Science Bulletin, 59(35):50665073, 2014.

[2] Weijing Song, LizheWang, Peng Liu, et al. Improved t-sne based manifold dimensional reduction for remote sensing data processing. Multimedia Tools and Applications, Feb 2018.

[3] Weijing Song, Lizhe Wang, Yang Xiang, et al. Geographic spatiotemporal big data correlation analysis via the HilbertHuang transformation. Journal of Computer and System Sciences, 89:130 (141), 2017.

[4] Robert E. Kennedy, Zhiqiang Yang, and Warren B. Cohen. Detecting trends in forest disturbance and recovery using yearly Landsat time series: 1. landtrendr — temporal segmentation algorithms. Remote Sensing of Environment, 114(12):2897 2910, 2010.

[5] Toshihiro Sakamoto, Nhan Van Nguyen, Akihiko Kotera, et al. Detecting temporal changes in the extent of annual ooding within the Cambodia and the Vietnamese Mekong Delta from modis time-series imagery. Remote Sensing of Environment, 109(3):295 313, 2007.

[6] Zhang L.F., Chen H., Sun X.J., et al. Designing spatial-temporal-spectral integrated storage structure of multi-dimensional remote sensing images. Journal of Remote Sensing, 21(1):62 73, 2017.

[7] Assis, Luiz Fernando, Gilberto Ribeiro, et al. Big data streaming for remote sensing time series analytics using map-reduce. In XVII Brazilian Symposium on GeoInformatics, 2016.

[8] D.B. Gonzalez and L.P. Gonzalez. Spatial data warehouses and solap using open-source tools. In 2013 XXXIX Latin American Computing Conference (CLEI), pages 112, Oct 2013.

[9] T.O. Ahmed. Spatial on-line analytical processing (solap): Overview and current trends. In 2008 International Conference on Advanced Computer Theory and Engineering, pages 10951099, Dec 2008.

[10] Lizhe Wang, Weijing Song, and Peng Liu. Link the remote sensing big data to the image features via wavelet transformation. Cluster Computing, 19(2):793810, Jun 2016.

[11] LizheWang, Jiabin Zhang, Peng Liu, et al. Spectral spatial multi-featurebased deep learning for hyperspectral remote sensing image classification. Soft Computing, 21(1):213221, Jan 2017.

[12] Weitao Chen, Xianju Li, Haixia He, et al. Assessing different feature sets' effects on land cover classification in complex surface-mined landscapes by Ziyuan-3 satellite imagery. Remote Sensing, 10(1), 2018.

[13] K.O. Asante, R.D. Macuacua, G.A. Artan, et al. Developing a ood monitoring system from remotely sensed data for the Limpopo basin. IEEE Transactions on Geo science and Remote Sensing, 45(6):17091714, June 2007.

[14] Matthew Rocklin. Dask: Parallel computation with blocked algorithms and task scheduling. In Kathryn Hu and James Bergstra, editors,Proceedings of the 14th Python in Science Conference, pages 130 136, 2015.

[15] Erik Thomsen. OLAP Solutions: Building Multidimensional Information Systems. John Wiley Sons, Inc., 2002.

[16] Z. Yijiang. The conceptual design on spatial data cube. In 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), pages 645648, April 2012.

[17] Sonia Rivest, Yvan Bedard, Marie-Josee Proulx, Martin Nadeau, Frederic Hubert, and Julien Pastor. Solap technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data. ISPRS Journal of Photogrammetry and Remote Sensing, 60(1):17 33, 2005.

[18] Matthew Scotch and Bambang Parmanto. Sovat: Spatial olap visualization and analysis tool. In Proceedings of the Hawaii International Conference on System Sciences, page 142.2, 2005.

[19] Junqing Fan, Jining Yan, Yan Ma, et al. Big data integration in remote sensing across a distributed metadata-based spatial infrastructure. Remote Sensing, 10(1), 2018.

[20] Jining Yan, Yan Ma, Lizhe Wang, et al. A cloud-based remote sensing data production system. Future Generation Computer Systems, 2017.

[21] Gilberto Camara, Luiz Fernando Assis, et al. Big earth observation data analytics: Matching requirements to system architectures. In Proceedings of the 5th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data, BigSpatial '16, pages 16, New York, NY, USA, 2016. ACM.

[22] SCIDB. A database management system designed for multidimensional data. http://scidb.sourceforge.net/project.html, 2017.

[23] Apache. Hadoop web site. http://hadoop.apache.org/, 2017.

[24] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, et al. The hadoop distributed le system. In IEEE Symposium on MASS Storage Systems and Technologies, pages 110, 2010.

[25] odc. Open data cube. http://datacube-core.readthedocs.io/en/latest/index.html, 2017.

[26] OpenStreetMap. the project that creates and distributes free geographic data for the world. http://www.openstreetmap.org, 2017.

[27] UCAR. Netcdf le format and api. http://www.unidata.ucar.edu/software/netcdf/, 2017.

[28] Stephan Hoyer and Joseph J.Hamman. Xarray: N-d labeled arrays and datasets in python. Journal of Open Research Software, 5(3), 2017.

[29] Weitao Chen, Xianju Li, Haixia He, et al. A review of ne-scale land use and land cover classification in open-pit mining areas by remote sensing techniques. Remote Sensing, 10(1), 2018.

[30] Xianju Li, Weitao Chen, Xinwen Cheng, et al. Comparison and integration of feature reduction methods for land cover classification with rapideye imagery. Multimedia Tools and Applications, 76(21):2304123057, Nov 2017.

[31] Xianju Li, Gang Chen, Jingyi Liu, et al. Effects of rapideye imagery's red-edge band and vegetation indices on land cover classification in an arid region. Chinese Geographical Science, 27(5):827835, Oct 2017.

[32] Jie Zhang, Jining Yan, Yan Ma, et al. Infrastructures and services for remote sensing data production management across multiple satellite data centers. Cluster Computing, 19(3):118, 2016.

[33] Ye Tian, Xiong Li, Arun Kumar Sangaiah, et al. Privacy-preserving scheme in social participatory sensing based on secure multi-party cooperation. Computer Communications, 119:167 178, 2018.

[34] Chen Chen, Xiaomin Liu, Tie Qiu, et al. Latency estimation based on trac density for video streaming in the internet of vehicles. Computer Communications, 111:176 186, 2017.