BIG DATA ANALYTICS FOR DETECTING CROP CULTIVATION USING CO-LOCATION MINING ALGORITHM

KALAISELVI.G, PRIYANGA.P, PRIYANGA .S, SANDHYA .R GUIDED BY R.DEEPAN (AP/CSE)

Abstract

Choose the required land and Check whether that land affected by any natural disaster or not.If the land was affect then fint the co-locations of the required land by using co-location mining algorithm.Find soil of that required land using sloping concept and produce the result.If there is no any disaster then produce the result using agriculture database.The assets of remote senses digital world daily gen- erate massive volume of real-time data (mainly referred to the term "Big Data"), where insight information has a potential signif- icif collected and aggregated effectively. In today's era, there is a great deal added to real-time remote sensing Big Data than it seems at first, and extracting the useful information in an efficient manner leads a system toward a major computational challenges, such as to analyze, aggregate, and store, where data are remotely collected. Keeping in view the above mentioned factors, there is a need for designing a system architecture that welcomes both real- time, as well as offline data processing. Therefore, in this paper, we propose real-time Big Data analytical architecture for remote sensing satellite application. The proposed architecture comprises three main units, such as 1) remote sensing Big Data acquisition unit(RSDU); 2) data processing unit (DPU); and 3) data anal- ysis decision unit (DADU). First, RSDU acquires data from the satellite and sends this data to the Base Station, where initial pro- cessing takes place. Second, DPU plays a vital role in architecture for efficient processing of real-time Big Data by providing filtra- tion, load balancing, and parallel processing. Third, DADU is the upper layer unit of the proposed architecture, which is responsible for compilation, storage of the results, and generation of decision based on the results received from DPU. The proposed architec- ture has the capability of dividing, load balancing, and parallel processing of only useful data. Thus, it results in efficiently ana- lyzing real-time remote sensing Big Data using earth observatory system. Furthermore, the proposed architecture has the capabil- ity of storing incoming raw data to perform offline analysis on largely stored dumps, when required. Finally, a detailed analysis of remotely sensed earth observatory Big Data for land and sea area are provided using Hadoop. In addition, various algorithms are proposed for each level of RSDU, DPU, and DADU to detect land as well as sea area to elaborate the working of an architecture. work was supported in part by the IT R&D Program of MSIP/IITP. [10041145], Self-Organized Software Platform (SoSp) for Welfare

Devices, in part by the Brain Korea 21 Plus Project (SW Human Resource Development Program for Supporting Smart Life) funded by Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea under Grant 21A20131600005. M. Mazhar Ullah Rathore, A. Paul, and A. Ahmad are with the School of Computer Science and Engineering, Kyungpook National University, Daegu 702-701, Korea (e-mail: rathoremazhar@gmail.com; paul.editor@gmail.com; awais.ahmad@live.com). B.-W. Chen is with the Department of Electrical Engineering, National Cheng

Kung University, Tainan 70101, Taiwan (e-mail: dennisbwc@gmail.com). B. Huang is with the Cooperative Institute of Meteorological Satellite Studies, Space Science and Engineering Center, University of Wisconsin- Madison, Madison, WI 53706 USA (e-mail: bormin@ssec.wisc.edu). W. Ji is with the Institute of Computing Technology, Chinese Academy of Science, Beijing 100190, China (e-mail: jwen@ict.ac.cn). Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org. Digital Object Identifier 10.1109/JSTARS.2015.2424683

Index Terms—Big Data, data analysis decision unit (DADU), data processing unit (DPU), land and sea area, offline, real-time, remote senses, remote sensing Big Data acquisition unit (RSDU).

I. INTRODUCTION R ECENTLY, a great deal of interest in the field of Big Data and its analysis has risen [1]–[3], mainly driven from extensive number of research challenges strappingly related to bonafide applications, such as modeling, processing, querying, mining, and distributing large-scale repositories. The term “Big Data” classifies specific kinds of data sets comprising form- less data, which dwell in data layer of technical computing applications [4] and the Web [5]. The data stored in the underly- ing layer of all these technical computing application scenarios have some precise individualities in common, such as 1) large- scale data, which refers to the size and the data warehouse; 2) scalability issues, which refer to the application’s likely to be running on large scale (e.g., Big Data); 3) sustain extrac- tion transformation loading (ETL) method from low, raw data to well thought-out data up to certain extent; and 4) develop- ment of uncomplicated interpretable analytical over Big Data warehouses with a view to deliver an intelligent and momen- tous knowledge for them [8]. Big Data are usually generated by online transaction, video/audio, email, number of clicks, logs, posts, social network data, scientific data, remote access sen- sory data, mobile phones, and their applications [6], [7]. These data are accumulated in databases that grow extraordinarily and become complicated to confine, form, store, manage, share, process, analyze, and visualize via typical database software tools. Advancement in Big Data sensing and computer technol- ogy revolutionizes the way remote data collected, processed, analyzed, and managed [9]–[12]. Particularly, most recently designed sensors used in the earth and planetary observatory system are generating continuous stream of data. Moreover, majority of work have been done in the various fields of remote sensory satellite image data, such as change detection [13], gradient-based edge detection [14], region similarity- based edge detection [15], and intensity gradient technique for efficient intraprediction [16]. In this paper, we referred the high- speed continuous stream of data or high volume offline data to “Big Data,” which is leading us to a new world of challenges [17]. Such consequences of transformation of remotely sensed data to the scientific understanding are a critical task [18], [34], [35]. Hence the rate at which volume of the remote access data

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

2 IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING

is increasing, a number of individual users as well as organi- zations are now demanding an efficient mechanism to collect, process, and analyze, and store these data and its resources. Big Data analysis is somehow a challenging task than locating, identifying, understanding, and citing data [19]. Having a large-scale data, all of this has to happen in a mechanized manner since it requires diverse data structure as well as semantics to be articulated in forms of computer-readable format. However, by analyzing simple data having one data set, a mechanism is required of how to design a database. There might be alternative ways to store all of the same information. In such conditions, the mentioned design might have an advantage over others for certain process and possible draw- backs for some other purposes. In order to address these needs, various analytical platforms have been provided by relational databases vendors [20]. These platforms come in various shapes from software only to analytical services that run in third-party hosted environment. In remote access networks, where the data source such as sensors can produce an overwhelming amount of raw data. We refer it to the first step, i.e., data acquisition, in which much of the data are of no interest that can be filtered or compressed by orders of magnitude. With a view to using such filters, they do not discard useful information. For instance, in consideration of new reports, is it adequate to keep that information that is mentioned with the company name? Alternatively, is it neces- sary that we may need the entire report, or simply a small piece around the mentioned name? The second challenge is by default generation of accurate metadata that describe the composition of data and the way it was collected and analyzed. Such kind of metadata is hard to analyze since we may need to know the source for each data in remote access. Normally, the data collected from remote areas are not in a format ready for analysis. Therefore, the second step refers us to data extraction, which drags out the useful information from the underlying sources and delivers it in a structured formation suitable for analysis. For instance, the data set is reduced to single-class label to facilitate analysis, even though the first thing that we used to think about Big Data as always describing the fact. However, this is far away from reality; sometimes we have to deal with erroneous data too, or some of the data might be imprecise. To address the aforementioned needs, this paper presents a remote sensing Big Data analytical architecture, which is used to analyze real time, as well as offline data. At first, the data are remotely preprocessed, which is then readable by the machines. Afterward, this useful information is transmitted to the Earth Base Station for further data processing. Earth Base Station performs two types of processing, such as processing of real-time and offline data. In case of the offline data, the data are transmit- ted to offline data-storage device. The incorporation of offline data-storage device helps in later usage of the data, whereas the real-time data is directly transmitted to the filtration and load balancer server, where filtration algorithm is employed, which extracts the useful information from the Big Data. On the other hand, the load balancer balances the processing power by equal distribution of the real-time data to the servers. The fil- tration and load-balancing server not only filters and balances the load, but it is also used to enhance the system efficiency.

Furthermore, the filtered data are then processed by the parallel servers and are sent to data aggregation unit (if required, they can store the processed data in the result storage device) for comparison purposes by the decision and analyzing server. The proposed architecture welcomes remote access sensory data as well as direct access network data (e.g., GPRS, 3G, xDSL, or WAN). The proposed architecture and the algorithms are imple- mented in Hadoop using MapReduce programming by applying remote sensing earth observatory data. This paper is organized as follows. In Section II, we give a detailed description about the motivation behind the proposed architecture. In Section III, we propose remote senses Big Data analytics architecture. In Section IV, we present a detailed analytics and discussion. Section V presented the results and evaluation of the system. Finally, Section VI offers conclusion and future directions of this paper.

II. MOTIVATION FOR REMOTE SENSING BIG DATA ANALYTICS

The increase in the data rates generated on the digital uni- verse is escalating exponentially. With a view in employing current tools and technologies to analyze and store, a massive volume of data are not up to the mark [21], since they are unable to extract required sample data sets. Therefore, we must design an architectural platform for analyzing both remote access real- time and offline data. When a business enterprise can pull-out all the useful information obtainable in the Big Data rather than a sample of its data set, in that case, it has an influential ben- efit over the market competitors. Big Data analytics helps us to gain insight and make better decisions. Therefore, with the intentions of using Big Data, modifications in paradigms are at utmost. To support our motivations, we have described some areas where Big Data can play an important role. Understanding environment requires massive amount of data collected from various sources, such as remote access satellite observing earth characteristics [measurement data set (MDS) of satellite data such as images], sensors monitoring air and water quality, metrological circumstances, and proportion of CO2 and other gases in air, and so on. Through relating all the informa- tion drifting such as CO2 emanation, increase or decrease on greenhouse effects and temperature, can be found out. In healthcare scenarios, medical practitioners gather massive volume of data about patients, medical history, medications, and other details. The above-mentioned data are accumulated in drug-manufacturing companies. The nature of these data is very complex, and sometimes the practitioners are unable to show a relationship with other information, which results in missing of important information. With a view in employ- ing advance analytic techniques for organizing and extracting useful information from Big Data results in personalized med- ication, the advance Big Data analytic techniques give insight into hereditarily causes of the disease.

III. REMOTE SENSING BIG DATA ANALYTICS ARCHITECTURE

The term Big Data covers diverse technologies same as cloud computing. The input of Big Data comes from social

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

RATHORE et al.: REAL-TIME BIG DATA ANALYTICAL ARCHITECTURE FOR REMOTE SENSING APPLICATION 3

Fig. 1. Remote sensing Big Data architecture.

networks (Facebook, Twitter, LinkedIn, etc.), Web servers, satellite imagery, sensory data, banking transactions, etc. Regardless of very recent emergence of Big Data architec- ture in scientific applications, numerous efforts toward Big Data analytics architecture can already be found in the literature. Among numerous others [23]–[27], we propose remote sens- ing Big Data architecture to analyze the Big Data in an efficient manner as shown in Fig. 1. Fig. 1 delineates n number of satel- lites that obtain the earth observatory Big Data images with sensors or conventional cameras through which sceneries are recorded using radiations. Special techniques are applied to process and interpret remote sensing imagery for the purpose of producing conventional maps, thematic maps, resource sur- veys, etc. We have divided remote sensing Big Data architecture into three parts, i.e., 1) remote sensing data acquisition unit (RSDU); 2) data processing unit (DPU); and 3) data analysis and decision unit (DADU). The functionalities and working of the said parts are described as below.

A. Remote Sensing Big Data Acquisition Unit (RSDU)

Remote sensing promotes the expansion of earth observa- tory system as cost-effective parallel data acquisition system to satisfy specific computational requirements. The Earth and Space Science Society originally approved this solution as the standard for parallel processing in this particular context [2]. As satellite instruments for Earth observation integrated more sophisticated qualifications for improved Big Data acqui- sition, soon it was recognized that traditional data processing technologies could not provide sufficient power for processing such kind of data. Therefore, the need for parallel process- ing of the massive volume of data was required, which could

efficiently analyze the Big Data. For that reason, the proposed RSDU is introduced in the remote sensing Big Data architec- ture that gathers the data from various satellites around the globe as shown in Fig. 2 [22]. It is possible that the received raw data are distorted by scattering and absorption by vari- ous atmospheric gasses and dust particles. We assume that the satellite can correct the erroneous data. However, to make the raw data into image format, the remote sensing satellite uses Doppler or SPECAN algorithms [28]. For effective data anal- ysis, remote sensing satellite preprocesses data under many situations to integrate the data from different sources, which not only decreases storage cost, but also improves analysis accuracy. Some relational data preprocessing techniques are data integration, data cleaning, and redundancy elimination [36]–[39]. After preprocessing phase, the collected data are transmitted to a ground station using downlink channel. This transmission is directly or via relay satellite with an appro- priate tracking antenna and communication link in a wireless atmosphere. The data must be corrected in different methods to remove distortions caused due to the motion of the platform relative to the earth, platform attitude, earth curvature, nonuni- formity of illumination, variations in sensor characteristics, etc. The data is then transmitted to Earth Base Station for further processing using direct communication link. We divided the data processing procedure into two steps, such as real-time Big Data processing and offline Big Data processing. In the case of offline data processing, the Earth Base Station transmits the data to the data center for storage. This data is then used for future analyses. However, in real-time data processing, the data are directly transmitted to the filtration and load balancer server (FLBS), since storing of incoming real-time data degrades the performance of real-time processing.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

4 IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING

Fig. 2. Remote sensing earth observatory image.

B. Data Processing Unit

In data processing unit (DPU), the filtration and load bal- ancer server have two basic responsibilities, such as filtration of data and load balancing of processing power. Filtration iden- tifies the useful data for analysis since it only allows useful information, whereas the rest of the data are blocked and are discarded. Hence, it results in enhancing the performance of the whole proposed system. Apparently, the load-balancing part of the server provides the facility of dividing the whole filtered data into parts and assign them to various processing servers. The filtration and load-balancing algorithm varies from anal- ysis to analysis; e.g., if there is only a need for analysis of sea wave and temperature data, the measurement of these described data is filtered out, and is segmented into parts. Each processing server has its algorithm implementation for processing incoming segment of data from FLBS. Each processing server makes statistical calculations, any measure- ments, and performs other mathematical or logical tasks to generate intermediate results against each segment of data. Since these servers perform tasks independently and in parallel, the performance proposed system is dramatically enhanced, and the results against each segment are generated in real time. The results generated by each server are then sent to the aggrega- tion server for compilation, organization, and storing for further processing.

C. Data Analysis and Decision Unit (DADU)

DADU contains three major portions, such as aggregation and compilation server, results storage server(s), and decision- making server. When the results are ready for compilation, the processing servers in DPU send the partial results to the aggregation and compilation server, since the aggregated results are not in organized and compiled form. Therefore, there is a need to aggregate the related results and organized them into a proper form for further processing and to store them. In the proposed architecture, aggregation and compilation server is supported by various algorithms that compile, organize, store, and transmit the results. Again, the algorithm varies from requirement to requirement and depends on the analysis needs. Aggregation server stores the compiled and organized results into the result's storage with the intention that any server can use it as it can process at any time. The aggregation server also sends the same copy of that result to the decision-making server to process that result for making decision. The decision-making server is supported by the decision algorithm, which inquire

Fig. 3. Flowchart of the remote sensing Big Data architecture.

different things from the result, and then make various decisions (e.g., in our analysis, we analyze land, sea, and ice, whereas other finding such as fire, storms, Tsunami, earthquake can also be found). The decision algorithm must be strong and cor- rect enough that efficiently produce results to discover hidden things and make decisions. The decision part of the architec- ture is significant since any small error in decision-making can degrade the efficiency of the whole analysis. DADU finally dis- plays or broadcasts the decisions, so that any application can utilize those decisions at real time to make their development. The applications can be any business software, general purpose community software, or other social networks that need those findings (i.e., decision-making). The self-explanatory flowchart supporting the working of the proposed architecture is depicted in Fig. 3.

IV. ANALYSIS AND DISCUSSION

Using the proposed architecture for offline as well online traffic, we perform a simple analysis on remote sensing earth observatory data. We assume that the data are big in nature and difficult to handle for a single server. The data are contin- uously coming from a satellite with high speed. Hence, special algorithms are needed to process, analyze, and make a decision from that Big Data. Here, in this section, we analyze remote sensing data for finding land, sea, or ice area. We have used the proposed architecture to perform analysis and proposed an algorithm for making decision. First, we take satellite-sensed Big Data samples from European satellite Agency (ESA) [22] to analyze land, sea, and ice separately. On the basis of these analyses, we proposed a set of algorithms for handling, pro- cessing, analyzing, and decision-making (detecting sea, land,

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

RATHORE et al.: REAL-TIME BIG DATA ANALYTICAL ARCHITECTURE FOR REMOTE SENSING APPLICATION 5

TABLE I DATA SETS INFORMATION

and ice area) for remote sensing Big Data images using our proposed architecture. In this section, we describe the data sets and tools that are used to perform analysis. Furthermore, we described the analysis findings of the data sets and proposed algorithms.

A. Tools, Data Set, and Implementation Environment

We used BEAM VISAT version 5.0 [29] and EnviView 2.8.1 [22] for simple analysis of the satellite data sets. Beam VISAT and EnviView provide an easy way to understand the structure of ENVISAT mission satellite data sets. They are also useful for simple statistical analysis. For complicated analysis and efficient processing of the Big Data sets, we could not use these tools. Apache Hadoop with MapReduce program using single node setup for sophisticated analysis is used for the implementa- tion of the proposed algorithm, since Hadoop provides the facility of parallel, high-performance computing using a large number of servers [30]. Therefore, it is suitable for analyzing a large amount of remote sensory image data. The proposed architecture uses a similar mechanism for load balancing; hence, preference is given to Hadoop for sophisticated analysis, algorithm development, and testing. We analyzed the ENVISAT mission data sets (e.g., products) with advanced synthetic apertures radar (ASAR) and medium-resolution imaging spectrometer (MERIS) instruments or sen- sor. The main focus is given to ENVISAT ASAR data sets because ENVISAT satellite mission has been continuously pro- viding global measurements for the earth including sea, land, ice, and forest since 2002 [28]. It has ten basic instrument data sets for sensing earth. However, we have considered only two instruments, i.e., ASAR and MERIS. More data sets are analyzed from MERIS, ASAR, and few from other ENVISAT sensors, but here we discussed the analytical results of five basic types of ASAR data sets taken from ESA [22].These five remote sensory imaging data sets covered different earth area including sea, desert, forest, beaches, and cities. The data sets included different earth area from Vietnam, Poland, and Germany, Western Sahara and Mauritania and Mali, South

Fig. 4. Data set-/product-covered area.

Africa, and Spain with different times as shown in Fig. 2. These products are of different types, i.e., altering polariza- tion medium-resolution image (APM), wide swath medium- resolution image (WSM), and global monitoring mode image (GMI). The software used for sensing data is ASAR with a different version of 4.02, 3.00S00, 5.03L03, 4.02. A detailed description of these products including their ID/name, product type, sensing time, software version, mis- sion, SPH descriptor and area covered is shown in Table I. The area covered corresponding to each product is also shown graphically using 2-D world map in Fig. 4.

B. Findings and Discussion

On Earth station, the reception of preprocessed and formatted data from satellite contains all or some of the following parts depending on the product. 1) Main product header (MPH): It includes the products basis information, i.e., id, measurement and sensing time, orbit, information, etc. 2) Special products head (SPH): It contains information spe- cific to each product or product group, i.e., number of data sets descriptors (DSD), directory of remaining data sets in the file, etc. 3) Annotation data sets (ADS): It contains information of quality, time tagged processing parameters, geo location tie points, solar, angles, etc. 4) Global annotation data sets (GADs): It contains calling factors, offsets, calibration information, etc.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

6 IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING

TABLE II DATA SETS BASIC STATISTICS

TABLE III STATISTICS OF PRODUCTS/DATA SETS FOR LAND AREA

TABLE IV STATISTICS OF PRODUCTS/DATA SETS FOR SEA AREA

5) Measurement data set (MDS): It contains measurements or graphical parameters calculated from the measurement including quality flag and the time tag measurement as well. The image data are also stored in this part and are the main element of our analysis. The MPH and SPH data are in ASCII format, whereas all the other data sets are in binary format. MDS, ADS, and GADs consist of the sequence of records and one or more fields of the data for each record. In our case, the MDS contains num- ber of records, and each record contains a number of fields. Each record of the MDS corresponds to one row of the satellite image, which is our main focus during analysis. In the basic analysis of all these products, we found that the satellite uses the SPECAN algorithm for processing raw data into image data. An image data are normally composed in rows and columns, i.e., matrices. In remote sensory Big Data, the MDS data sets containing the image data are in the form of records. Number of records shows the number of rows in the satellite image and number of sample/line shows the number of field/column in the image. The value against record (i) and sample (j) corresponds to the pixels value of the row (i) and column (j) of the image. Table II shows the algorithm used by the satellite (for processing raw data to image), the number of records in each data sets, number of samples in each record, and mean and standard deviation (SD) of all the MDS pro- cess data values corresponding to all records and samples. It is noticed that the mean value of the product is quite lower than all other products mean value. The products 2 only covered the land area; hence, their mean value is quite lower, whereas all other products covered both land and sea area. Since in land satellite images, the color of the land particles is mostly nearer

to the block. Therefore, their value is quite lower. Hence, the overall mean value for image particles is lower. In the analysis of the products, we consider 20 random blocks image data, where each block contains 20 000–30 000 sample image-related values from MDS records for both land and sea area. We calculated the mean and SD of all image sample values in each block and calculated the maximum sample value in the block and the normal trend of sample values. We also observed the distribution of MDS image values for land area as well as sea area and then tried to find out the major difference between their data. We assume that the mean value for the land area should be lower as compared to sea data. The land normally has greenery (except deserts), and other objects whose color is nearer to black. Hence, the pixel value is lower, which results in reducing the overall mean value. The pixels SD for the land data is also higher in case of land area. Normally, sea has one color, and there are very few particles on the surface of the sea, which can be ignored. As a result, the color of the sea remains almost same. Therefore, the SD for the sea data is lower and for land, SD is higher as the land has many different things with different colors on its surface. Accordingly, we have considered mean, SD, and the maximum value as the basic parameters for our analysis. Tables III and IV show the overall statistical analytical results of all the products with respect to land area and sea area, respectively, in which the minimum and maximum values of mean and SD among all the blocks are presented. The analysis findings are almost according to our assumption except for few abnormalities. It can also be seen that the mean values for land areas are quite lower as compared to sea area and SD values in case of land are quite higher as compared to sea area.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

RATHORE et al.: REAL-TIME BIG DATA ANALYTICAL ARCHITECTURE FOR REMOTE SENSING APPLICATION 7

Fig. 5. (a) PVD of Product 1 for land area. (b) PVD of Product 2 for land area. (c) PVD of Product 3 for land area. (d) PVD of Product 4 for land area. (e) PVD of Product 5 for land area.

Fig. 6. (a) PVD of Product 1 for sea area. (b) PVD of Product 2 for sea area. (c) PVD of Product 4 for sea area. (d) PVD of Product 5 for sea area.

In our analysis, some abnormalities can be found, i.e., the SD of the product 2 for land is quite lower unexpectedly as in Table III. This is only because of lower values of each sample value of MDS record. Therefore, the mean is quite lower, and the resultant SD is also lower. However, the absolute difference between mean and the SD is still higher. In Table III, some other abnormalities can also be found, i.e., the maximum SD is quite higher for product 3 for sea data. This happens quite rarely only when there is some noise added to the data or when the color of

the sea changes due to some forest shade, port/beach, etc. We also found that the mean value for Product 4 is lower due to the color of the sea. Hence, the SD remains lower in both of these cases as per our assumptions. Our sample values distribution analysis is shown in Figs. 5 and 6 with respect to land and sea, respectively. There is a major difference between both of these distributions. In Figs. 5 and 6, few sample values are greater than 2000 for land data in all the products. There is additional diversity,

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

8 IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVssATIONS AND REMOTE SENSING

variability in values results in increasing the SD for land data. In case, if there is large lake in between the land, then the min- imum variability is noticed that results in minimizing the SD of that block, i.e., from sample index 700 to 800 as shown in Fig. 5(b). As shown in Fig. 6, it is obvious that the variation in sample values of sea data is lower and most of the values are higher than 2500, which clearly differentiate sea from land area as in Fig. 6(a) and (b). Some values of sea area in products 4 and 5 are quite lower than 1000.

C. Algorithm Design and Testing

On the basis of the analysis made in the previous section, a set of algorithms is proposed to process high-speed, large amount of real-time remote sensory image data using our pro- posed architecture. It works on both DPU and DADU by taking data from satellite as input to identify land and sea area from the data set. The set of algorithms contains four simple algorithms, i.e., algorithm I, algorithm II, algorithm III, and algorithm IV that work on filtrations and load balancer, pro- cessing servers, aggregation server, and on decision-making server, respectively. Algorithm I, i.e., filtration and load balancer algorithm (FLBA) works on filtration and load balancer to filter only the require data by discarding all other information. It also provides load balancing by dividing the data into fixed size blocks and sending them to the processing server, i.e., one or more distinct blocks to each server. This filtration, dividing, and load-balancing task speeds up our performance by neglect- ing unnecessary data and by providing parallel processing. Algorithm II, i.e., processing and calculation algorithm (PCA) processes filtered data and is implemented on each processing server. It provides various parameter calculations that are used in the decision-making process. The parameters calculations results are then sent to aggregation server forfurther process- ing. Algorithm III, i.e., aggregation and compilations algorithm (ACA) stores, compiles, and organizes the results, which can be used by decision-making server for land and sea area detection. Algorithm IV, i.e., decision-making algorithm (DMA) identi- fies land area and sea area by comparing the parameters results, i.e., from aggregation servers, with threshold values.

Algorithms parameters and variables: Following are the parameters and variables used in the proposed algorithms. B1, B2, B3, B4, B5 ... BN is image fixed size blocks. XBi: Mean of sample values of block Bi, where i={1,2,3, 4...N}. XBi= Sum of all values of the block Bi/size of block =BS j=1 Vj BS Vj: jth value in block Bi. 1 ≤ j ≤ BS. SDBi: Standard deviation of sample values of block Bi. SDBi =______________________________

BS j=1 (VJ −XBi) BS . Abs_Diff: Absolute difference between XBi and SDBi. AbsDiff = XBi−SDBi . Maxval: Threshold value is set and is greater than normal range to check how many values of the block are deviated from the normal range. NGmaxval: Number of values in the block is greater than Maxval. Below are the threshold variables, which are set on the basis of analysis, i.e., ∂X: Mean threshold is set on the basis of analysis, which is used to compare the mean value of each block with threshold for detecting land, sea, or any other area. ∂SD: SD threshold is set on the basis of analysis, which is used to compare the SD value of each block with threshold for detecting land, sea, or any other area. ∂Abs_diff: Absolute difference threshold is set on the basis of analysis, which is used to compare absolute value of each block with threshold for detecting land, sea, or any other area. ∂NGmaxval: Threshold for number of values that are greater than Maxval is set on the basis of analysis, which is used to compare NGmaxval value of each block with threshold for detecting land, sea area, or any other area.

Algorithm I. Filtration and Load Balancing Algorithm (FLBA)

Input: Satellite process data set/product Output: filtered Image data in fixed size block and send each block to processing server Steps: 1. Filter Image related data i.e. Processed data in MDS. All other unnecessary data will be discarded. 2. Divide the image into fixed size block i.e. BS = 100×100 MDS process_data values, row by row fashion or column by column. Each block will be denoted by Bi where 1 ≤ i ≤ BS 3. Make two samples of blocks so that only half of the part is processed. i.e., PSB = {B1,B3,B5,...,BN−1} and UPSB = {B2,B4, B6,B8,...,BN} 4. Transmit UPSB directly to aggregation server without pro- cessing. 5. Assign and transmit each distinct block(s) Bi of PSB to various processing servers in DPU.

Description: This algorithm takes satellite data or product and then filters and divides them into segments and performs load-balancing algorithm. In step 1, the image-related data resides in the MDS part of the product, is filtered out. In step 2,

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

RATHORE et al.: REAL-TIME BIG DATA ANALYTICAL ARCHITECTURE FOR REMOTE SENSING APPLICATION 9

filtered data are the association of different numbers of record and each record is different numbers of sample, which results in forming a matrix. The matrix is then divided into fixed sized blocks. A set of building blocks is {B1,B2,B3,B4,...,BN} that contains 100×100 values. Block size can be more or less depending on the number of servers and computation power. In step 3, these blocks are divided into two parts, i.e., PSB and UPSB. UPSB is processed by PDU, whereas the decision of UPSB is based on PSB processing results.

Algorithm II. Processing and Calculation Algorithm (PCA)

Input: Block Bi Output: statistical parameters results and transmit them to aggregation server. Steps: 1. For each Block Bi, Calculate a. XBi b. S.DBi c. Abs_Diff d. NGmaxval 2. Transmit the results against block id and product id to the aggregation server in DADU

Description: The processing algorithm calculates results for different parameters against each incoming block and sends them to the next level. In step 1, the calculation of mean, SD, absolute difference, and the number of values, which are greater than the maxi- mum threshold, are performed. Furthermore, in the next step, the results are transmitted to the aggregation server.

Algorithm III. Aggregation and Compilation Algorithm (ACA) Input: Block Bi results Output: compiling, storing and sending PSB results and UPSB blocks information to decision-making server. Steps: 1. Collect Every Bi's result of PSB 2. Compile them and transmit them to Decision-making server. 3. Store PSB blocks with results and UPSB blocks without result into RBMS in result storage.

Description: ACA collects the results from each process- ing servers against each Bi and then combines, organizes, and stores these results in RDBMS database. It also stores UPBS information into the database. ACA transmits a copy of PSB results and UPSB information to decision-making server for real-time decision-making.

Algorithm IV. Decision-making algorithm (DMA)

Input: PSB results and UPSB information Output: each block Bi with decision, land block or sea. Finally, the whole image is divided into sea and land area Rules: Following rules are made on the basis of land area analysis discussed in Section III for detecting land block 1. $X\ Bi \leq \partial X$ 2. $S.DBi \geq \partial S.D$

3. Abs_Diff≥ ∂Abs_diff 4. NGmaxval ≤ ∂NGmaxval

Steps: 1. For Each (Bi of PBS) {

If (Rule1 ==trueandRule2 ==true) Status_Bi = Land Else if (Rule1 ==falseandRule2 ==false) Status_Bi = Sea Else { If (Rule3 ==falseandRule4 ==false)) Status_Bi = Sea Else Status_Bi = Land }

} 2. For Each (Bi of UPBS)

{

If (Status_Bi−1 == Landand status_Bi+1 = Land) Status_Bi = Land Else If (Status_Bi−1 == Seaand status_Bi+1 = Sea) Status_Bi = Sea Else Status_Bi =! (Status_Bi−1 ⨁status_Bi+1 ⨁status_Bi+3) }

Description: DMA takes results of each block of PSB and UPSB information as well. It then analyzes results of each block of PSB and determines whether the block belongs to land or sea. For each block Bi of PSB if $XBi \leq \partial X$ and $SDBi \geq \partial SD$, then the Bi is detected as land block and if $XBi \leq \partial X$ and $SDBi \geq \partial SD$ both are false then the block Bi is detected as sea. If $XBi \leq \partial X$ is false and $SDBi \geq \partial SD$ is true or vice versa, then the block is tested for rule 3 and 4, if both rules are false, then it is detected as sea and otherwise, it is detected as land. For UPSB, the block is identified on the basis of neighbor's block results. The neighbor block of each UPSB is in PSB. For every UPSB block Bi, if their neighbors blocks, i.e., Bi+1, Bi−1 that are PSB blocks, are land, then Bi is detected as land. If its neighbor blocks are sea, then Bi is detected as sea. For its neighboring blocks, if one is considered as land and other is considered as sea, then we need to find the decision of next neighbor, i.e., Bi+3, Bi is detected as land, if two of these neighbor block are land; otherwise, it will be sea. Finally, when each block of PSB and UPSB is detected as sea or land, then the decision-making process is completed, which uses these decisions in any application or just for announcement or display.

V. RESULTS AND IMPLEMENTATION

We implemented our algorithms in simple java language using Beam-5.0 library [29] as well in Hadoop using

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

10 IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING

TABLE V ACCURACY RESULTS OF LAND AND SEA DETECTION ALGORITHM

Fig. 7. (a) Product 1 resulted image. (b) Product 3 resulted in image. (c) Product 4resulted image. (d) Product 5 resulted image.

MapReduce, initially in a single-node environment. In the Hadoop implementation, Map function takes the image block offset as a key and the image block (pixel values) as a value parameter. Since Hadoop

MapReduce cannot directly process image blocks, the whole product image data are converted into sequence file to be processed using MapReduce. In such a way, one line of the sequence file contains one image block. Map function performs parameters calculations on incoming block values and finally sends the block number as a key and list of parameters results as a value to the Reduce function. Reduce function uses parameter results for performing decision-making on them. We test and evaluate our algorithms with respect to accuracy and processing time using various ESA products [22]. Accuracy evaluation is done by considering two parameters, such as true positive (TP), which shows the measurements in percentage

Fig. 8. Average processing time of products using Hadoop implementation.

Fig. 9. Efficiency comparison of Hadoop Map Reduce implementation and simple Java implementation.

(%) the land block are correctly identified and false positive (FP) shows the measurements in percentage (%), the sea blocks are incorrectly identified as land blocks. Our proposed system detects different types of area of the world, such as land and sea with the overall accuracy of more than 95% TP and less than 3% FP. Product 2 does not have sea area. Therefore, all blocks are detected as land that results in 100% TP and 0% FP. The accuracy results are shown in Table V. Fig. 7(a)–(d) delineates graphical result of detection in which the land and sea area are detected and separated by the algorithms used in the proposed architecture in product 1, product 3, product 4, and product 5, respectively. Efficiency measurements are taken by considering the aver- age processing time to process 1-MB data of various products. MapReduce implementation of the analysis algorithm takes less than 1 s the average processing time for various products except Product 3, which takes 1.5 s the average processing time. This processing time among various products varies due to the usage of different bands and image modes, depending on product type. The average processing time for various products is shown in Fig. 8. Finally, a comparison is made between the Hadoop MarReduce implementation and the simple Java implementa- tion of the proposed algorithms using average processing time measurements. The graph shown in Fig. 9 makes the compari- son obvious. For small-size products, i.e., less than 200 MB, the Hadoop implementation takes more average processing time to

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

RATHORE et al.: REAL-TIME BIG DATA ANALYTICAL ARCHITECTURE FOR REMOTE SENSING APPLICATION 11

process 1 MB data of the product, while a simple Java imple- mentation is efficient in this case. However, when the product size is increasing, the average process time starts decreasing in MapReduce implementation. Moreover, when the product size exceeds 200 MB, it produces better results as compared with simple Java implementation. Hence, for smaller size products, the Hadoop implementation is not efficient because of its lots of input and output operations due to Map and

Reduce function. In the case of large-size products, Hadoop divided whole prod- ucts into blocks and performed parallel tasking on them, which resulted in increasing efficiency.

VI. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we proposed architecture for real-time Big Data analysis for remote sensing application. The proposed archi- tecture efficiently processed and analyzed real-time and offline remote sensing Big Data for decision-making. The proposed architecture is composed of three major units, such as 1) RSDU; 2) DPU; and 3) DADU. These units implement algorithms for each level of the architecture depending on the required anal- ysis. The architecture of real-time Big is generic (application independent) that is used for any type of remote sensing Big Data analysis. Furthermore, the capabilities of filtering, divid- ing, and parallel processing of only useful information are performed by discarding all other extra data. These processes make a better choice for real-time remote sensing Big Data analysis. The algorithms proposed in this paper for each unit and subunits are used to analyze remote sensing data sets, which helps in better understanding of land and sea area. The pro- posed architecture welcomes researchers and organizations for any type of remote sensory Big Data analysis by developing algorithms for each level of the architecture depending on their analysis requirement. For future work, we are planning to extend the proposed architecture to make it compatible for Big Data analysis for all applications, e.g., sensors and social networking. We are also planning to use the proposed architecture to perform complex analysis on earth observatory data for decision making at real- time, such as earthquake prediction, Tsunami prediction, fire detection, etc.

REFERENCES

[1] D. Agrawal, S. Das, and A. E. Abbadi, “Big Data and cloud comput- ing: Current state and future opportunities,” in Proc. Int. Conf. Extending Database Technol. (EDBT), 2011, pp. 530–533.

[2] J. Cohen, B. Dolan, M. Dunlap, J. M. Hellerstein, and C. Welton, “Mad skills: New analysis practices for Big Data,” PVLDB, vol. 2, no. 2, pp. 1481–1492, 2009.

[3] J. Dean and S. Ghemawat, “Mapreduce: Simplified data processing on large clusters,” Commun. ACM, vol. 51, no. 1, pp. 107–113, 2008.

[4] H. Herodotou et al., “Starfish: A self-tuning system for Big Data ana- lytics,” in Proc. 5th Int. Conf. Innovative Data Syst. Res. (CIDR), 2011, pp. 261–272.

[5] K. Michael and K. W. Miller, “Big Data: New opportunities and new challenges [guest editors’ introduction],” IEEE Comput., vol. 46, no. 6, pp. 22–24, Jun. 2013.

[6] C. Eaton, D. Deroos, T. Deutsch, G. Lapis, and P. C. Zikopoulos, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. New York, NY, USA: Mc Graw-Hill, 2012.

[7] R. D. Schneider, Hadoop for Dummies Special Edition. Hoboken, NJ, USA: Wiley, 2012.

[8] A. Cuzzocrea, D. Saccà, and J. D. Ullman, “Big Data: A research agenda,” in Proc. Int. Database Eng. Appl. Symp. (IDEAS’13), Barcelona, Spain, Oct. 09–11, 2013

[9] R. A. Schowengerdt, Remote Sensing: Models and Methods for Image Processing, 2nd ed. New York, NY, USA: Academic Press, 1997.

[10] D. A. Landgrebe, Signal Theory Methods in Multispectral Remote Sensing. Hoboken, NJ, USA: Wiley, 2003.

[11] C.-I. Chang, Hyperspectral Imaging: Techniques for Spectral Detection and Classification. Norwell, MA, USA: Kluwer, 2003.

[12] J. A. Richards and X. Jia, Remote Sensing Digital Image Analysis: An Introduction. New York, NY, USA: Springer, 2006.

[13] J. Shi, J. Wu, A. Paul, L. Jiao, and M. Gong, “Change detection in synthetic aperture radar image based on fuzzy active contour mod- els and genetic algorithms,” Math. Prob. Eng., vol. 2014, 15 pp., Apr. 2014.

[14] A. Paul, J. Wu, J.-F. Yang, and J. Jeong, “Gradient-based edge detection for motion estimation in H.264/AVC,” IET Image Process., vol. 5, no. 4, pp. 323–327, Jun. 2011.