Enhancing big data warehousing and analytics for spatio-temporal massive data

Hanen Balti
Riadi Laboratory/University of Manouba
Manouba, Tunisia *hanen.balti@ensi-uma.tn*

Nedra Mellouli
LIASD Laboratory/ University of Paris8
Paris, France *n.mellouli@iut.univ-paris8.fr*

Ali Ben Abbes
Riadi Laboratory/University of Manouba
Manouba, Tunisia *ali.benabbes@yahoo.fr*

Imed Riadh Farah
Riadi Laboratory/University of Manouba
Manouba, Tunisia *imedriadh.farah@isamm.uma.tn*

Yanfang Sang
Key Laboratory of Water Cycle and Related Land Surface Processes/Institute of Geographic Sciences and Natural Resources Research/Chinese Academy of Sciences
Beinjing, China *sangyf@igsnrr.ac.cn*

Myriam Lamolle
LIASD Laboratory/ University of Paris8
Paris, France *m.lamolle@iut.univ-paris8.fr*

Abstract

The increasing amount of data generated by earth observation missions like Copernicus, NASA Earth Data, and climate stations is overwhelming. Every day, terabytes of data are collected from these resources for different environment applications. Thus, this massive amount of data should be effectively managed and processed to support decision-makers. In this paper, we propose an information system-based on a low latency spatio-temporal data warehouse which aims to improve drought monitoring analytics and to support the decision-making process. The proposed framework consists of 4 main modules: (1) data collection, (2) data preprocessing, (3) data loading and storage, and (4) the visualization and interpretation module. The used data are multi-source and heterogeneous collected from various sources like remote sensing sensors, biophysical sensors, and climate sensors. Hence, this allows us to study drought in different dimensions. Experiments were carried out on a real case of drought monitoring in China between 2000 and 2020.

Keywords: Big data analytics, Data warehouse, Storage, Spatio-temporal, Hive, Disaster management, Drought

1. Introduction

Earth data are rich. They preserve several facets of the data such as temporal and spatial features. These data are generally managed by Geographic Information System (GIS). A GIS is mainly defined as a computer system for collecting, storing, querying, analyzing, and dis-

playing geospatial data [3]. GIS was impacted by the rapid and ongoing growth of big earth data. Big Earth data are based around Earth Sciences and primarily include remote sensing data, weather data, biophysical data, atmospheric data, human-derived activities, and many others [10]. Big Earth data is characterized as being massive, noisy, multi-source, heterogeneous, multi-temporal, multi-scalar, highly dimensional, highly complex, non-stationary, and a mixture of structured and unstructured. It consists of all data related to the Earth, including the Earth's interior, surface, atmosphere, and near-space environment. Big Earth data are characterized by 4 Vs.: the volume refers to the amount of collected data more than 50, 000 TeraBytes (TB) in 2015 and will reach 350.000 TB by 2030 [14], the velocity refers to the rate at which new data are generated or the rate at which data is processed, the variety refers to the Earth big data, and the veracity refers to the overall quality of the available data. The quality of data can be impacted by noise or abnormalities in the original data gathering process [14, 1]. Hence, designing and implementing an efficient and sustainable data warehouse (DW) for disaster management is a critical issue. It defines a standardized data representation through its schema model and stores the multiple datasets so that they can be analyzed to extract relevant knowledge. There are three possible models to organize the data stored in a Data Warehouse: star, snowflake, and constellation modeling. These models are mainly composed of facts and dimensions. The fact table helps the user analyze the dimensions of the problem, which helps in making decisions to improve their results. Moreover, the dimension tables help to bring together the dimensions with which measurements should be taken [4]. The key difference between the fact table and dimension table is that the latter contains attributes with which actions are taken in the fact table. Therefore, star modeling is the simplest model. It consists of one fact table and several dimension tables [5]. The snowflake schema is a type of star schema that includes the hierarchical shape of dimensional tables [4]. In this model, there is a fact table made up of different dimensions and sub-dimension tables linked by primary and foreign keys to the fact table. Splitting helps reduce redundancy and prevents memory loss. A snowflake diagram is easier to manage but complex to design and understand. The fact constellation schema has multiple fact tables sharing dimension tables. This model is more complex than the star and snowflake models. Despite the advantages that the DW gives, there is a lack of reports in the literature that focus on DW design with the view to enable Disaster management Big Data analytics and mining. The design of large-scale DWs is challenging, as the earth observation data is spatial, temporal, complex, heterogeneous, high dimensional, and collected from multi-sources. Hence, the earth observation data sources are much diversified and have different levels of quality. This paper addresses some issues in Big Data Warehousing systems for disaster management using massive heterogeneous earth data such as Spatio-temporal querying of drought data.

Traditional DW conceptual models can be summarized in two approaches: Inmon and Kimball. Inmon offers an integrated data solution with a unified source whose major improvement is to overcome data redundancy. However, the complexity of the model increases over time as more tables are added to the data model. Not to mention that this approach requires experts to effectively manage a data warehouse. Using the Kimball architecture, the data is not integrated, which can very quickly cause irregularities between the data updated in the Kimball DW architecture and its sources. Indeed, in the data warehouse, techniques for denormalizing redundant data are added to the database tables. In addition, performance problems can occur due to the addition of columns in the fact table because of their extended dimensions. Inmon and Kimball data warehouse concepts can be used in a hybrid manner to successfully design data warehouse data models. In this paper we propose hybrid concepts where the major contributions in this paper could be resumed around 1) building a low latency spatio-temporal big data warehouse based on structured and unstructured data; 2) proposing a heterogeneous data loading module to efficiently load data into Hadoop. This module parallel loads the data to improve quick data

ingestion; 3) providing statistical analysis through SQL-like queries; 4) providing advanced interpretations for decision-making support. This paper is organized as follows: in Section 2, we present the state of the art of big data and data warehouses used in several fields. In Section 3, we present the proposed methodology. In Section 4 we present the experimentation, results, and discussion. We conclude this work in Section 5.

2. Related work

In the literature, several works used big data and data warehouses in different domains. [12] proposed a continental level agricultural data warehouse. They used 29 Crop datasets, and evaluate the performance of the proposed agricultural data warehouse and present some queries to extract knowledge about the management of crops. [8] proposed an agricultural data integration method using a constellation schema that is designed to be flexible enough to incorporate other datasets and big data models. They extracted knowledge with the view to improve crop yield; these include finding suitable quantities of soil properties, herbicides, and insecticides for both increasing crop yield and protecting the environment. [9] proposed a layering DW model, giving an ETL process for integrating the satellite data and designing two types of the application program interface. This study proved that a data warehouse is an effective solution for storing and mining big data earth observation satellites. In [16], the authors extended the entity-relationship model and proposed the multi-dimensional entity-relationship model to model operational and analytical data. They presented new representation elements and provided the extension of an analytical schema. The proposed model intends to handle cow data (milking, housing, disease, feeding, etc). [17] aims to present a big data warehousing architecture that can adjust to user needs and requirements as well as updates in the underlying data sources automatically or semi-automatically.

Unlike traditional data warehouses, modern data warehousing solutions automate the repetitive tasks involved in designing, developing, and deploying a data warehouse design to meet evolving user's business requirements. Hadoop eco-system and Oracle still seducing big data analysts in different fields as they are efficient in managing the voluminous amount of data. However, earth observation big data are characterized by their high complexity. These data have a high Spatio-temporal resolution, as they are gathered from different sources and are heterogeneous, and their management demands effective solutions. Hence, the present paper aims to load, store, manage, and interpret data and knowledge derived from the proposed framework and validate it by experts. Thus, this paper aims to design and deploy a low latency big data warehouse for drought monitoring. This data warehouse is scalable; it can support the continuously growing volume of data, data heterogeneity, and the management of environment applications data.

3. Methodology

This methodology aims to integrate large-scale heterogeneous big data from multiple sources (e.g. Climate data, remote sensing data, hydrological data) into a big data warehouse to provide decision support to effectively prevent droughts.
These data are defined by big data dimension (i.e. Volume, Variety, Veracity, and Velocity), as they are gathered from various resources. The proposed methodology consists of 4 main modules: The first module is data collection, the second module is data preprocessing, then comes the data loading and storage and finally we have the visualization and decision-making module. Fig.1 represents the workflow of the proposed architecture.

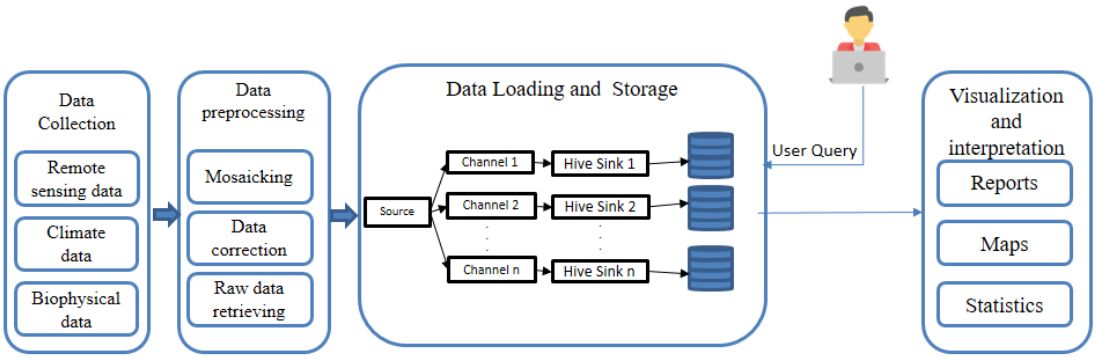

Fig. 1. The proposed architecture

3.1. Data collection

Data collection consists of generating and gathering data from different resources [1]. Drought is the interaction of several types of factors such as remote sensing data, climate data, and biophysical data [11]. These data are massive and heterogeneous. The collected data are remote sensing data (.e.g. Normalized Difference Vegetation Index (NDVI), Land Surface Temperature (LST)), Climate data (.e.g. Standardized Precipitation Evapotranspiration Index (SPEI), Evapotranspiration (ETP), humidity, precipitation, wind speed, pressure), biophysical data (e.g. Soil Moisture). These data are a range of structured, semi-structured, or unstructured data they are also characterized by their multidimensionality (e.g. multi-spectral, multi-resolution, multi-temporal data). This layer encompasses different data sources relevant to earth observations and deals with different data format types (i.e: NetCDF, CSV, hdr). Monthly, the data volume increases considerably. Thus, every day, Gigabytes of data are generated from different sources. For example, the remote sensing data (.i.e. NDVI, LST) are formatted in .hdr, the TRMM data, and the biophysical data are formatted in NetCDF and the climate data are formatted in .csv. The variety of data sources presented in this layer point to the heterogeneity related to their software applications and the used storage systems. Table 3 describes the data quantity and temporality.

Table 1. Data description

Data	Spatial Resolution	Temporal Resolution	Volume	Variety	Veracity	Velocity
LST	1km	8 days	x	-	x	x
NDVI	1-km	16 days	x	-	x	x
Precipitation	0.25°x0.25°	Daily	-	-	-	x
Climate Data	-	Daily	-	x	x	x
Soil Moisture	0.25°x0.25°	Monthly	-	-	-	x

3.2. Data preprocessing

The collected data are gathered from different sources. Hence, they contain different types of imperfections. For example, satellite images may have geometric distortion and atmospheric noises. Climate data can include some erroneous values. Thus, it is important to perform different operations on these data to improve their quality by applying several operations such as mosaicking, data correction, and raw data retrieving [1]. The mosaicking operation is applied to the remote sensing data. The collected images are a massive amount of tiles covering the study area. The purpose of this operation is to reconstruct one satellite image composed of many strips. The data correction operation consists of three different types of correction. The geo-

metric correction consists of avoiding geometric distortions from a distorted image, and is done by establishing the relationship between the image coordinate system and the geographic coordinate system using calibration data of the sensor, the atmospheric correction is to retrieve the surface reflection that characterizes the surface properties from remote sensing data by removing the atmospheric effects and the value correction which consists of correcting the erroneous values or identifying the missing values. Finally, the raw data retrieving consists of extracting valuable information from different sources of information such as calculating the NDVI, the LST, the SPEI, and ETP.

3.3. Data Loading and data Storage

Data loading

Data loading consists of transferring data into the Hadoop system. Drought data are gathered from different sources, therefore, they may be structured, semi-structured, and unstructured data. These data are loaded from different sources; and will be ingested by flume into HDFS. Data loading is performed using the Map-Only algorithm. The data is stripped into splits of uniform size. This method is called Block size-oriented storage. This ingestion method considers fault tolerance constraint as it performs several replicas of the chunks of data. For real-time data ingestion, Apache flume is used. This tool ingests online streaming semi-structured and unstructured data in HDFS. The flume agent is composed of 3 main components:
The source accepts the data from an incoming stream source, the channel is local temporary storage between the source and the HDFS and the sink collects data from the channel and commits it to the HDFS. In this work, we propose a distributed loading task using the Flume agent. The mapping job in Flume is performed on the channels. Each channel contains a strip of the data.

Data storage

The data warehouse is based on a simple and effective design for big earth observation data analysis as a multidimensional model. In this paper, a snowflake schema is proposed. The snowflake schema presents more details about the data than the star schema. It provides the ability to use more complex queries that means that it supports powerful analytics and many-to-many relationships. Fig.2 is composed of one Fact table and 13 dimension tables D=(Product_Dimension, Sensor_Dimension, Image_Dimension, SatelliteFeature_Dimension, Drought_Index_Dimension, ClimateStation_Dimension, Date_Dimension, ClimateFeature_Dimension, BiophysicalFeature_Dimension, BiophysicalStation_Dimension, Location_Dimension, Country, Province).
The DW is presented by: (F, D{}, HD_i{}) where:
-F: the Fact table
-D: {D_1, ..., D_n}: refers to the dimensions defined below, with n is the number of dimension tables
- HD_i{}: refers to the hierarchies for each dimension Di defined by HD_i={h_1, ..., h_k} with k is the number of hierarchies for each dimension.
The fact table: Each fact is defined by F: (NameF, M{}) where:
- NameF: is the name of the fact
-M{} = (m_1, ..., m_n) refers to the measures
The dimension table:
A dimension is defined by D: (NameD, A{}, H{}, TypeD) where:
- NameD: name of dimension
- A{}=(a_1, ..., a_l) is a set of attributes
- H{}=(h_1, ..., h_z) is a set of hierarchies

- TypeD $\subset$ [T, S]: a dimension could be temporal or spatial dimension.

The measure:

A measure M is defined by M:(NameM, TypeM, FuncM) where:

- NameM: name of the measure
- TypeM: the type of the measure
- FuncM: set of aggregation functions compatible with summarization property of the measure where $FuncM \subset \{SUM, AVG, MAX, MIN...\}$.

In our case, the fact table is named OperationFact, Sensor_Dimension is an example of the dimension tables and the measure is for example AVG_TEMP(). To mine the data stored in the proposed DWH, HiveQL and ElasticSearch are used. They provide an SQL-type environment to deal with tables, databases, and queries.

3.4. Visualization and interpretation

Decision-makers and scientists need to understand drought phenomena. Thus, they need to use big data to further develop the traditional decision-making process. Therefore, this module aims to present the final results in form of representations that help them to understand and deduce potential insights. To dialog, the data stored in the previous module, users, and decision-makers need to interrogate the DWH to extract valuable information for decision making. Using Apache Hive, various queries are provided such as data modeling by the creation of dimensions and facts, ETL functionalities like Extraction, Transformation, and loading data, and a faster-querying tool using Hadoop. Several representations are used for visualization such as charts, maps, and textual reports. The purpose of these representations is to show and discuss the trends, the Spatio-temporal variability, and the intensity of drought in a given region.

4. Experimentation and validation

To validate our methodology, the study area is presented in this section, and the implementation of the big data warehouse architecture is described and finally, some results are interpreted and discussed.

4.1. Study Area

China is situated on the east side of ASIA, and it has quite a lot of topography conditions. China's numerous land-forms and geographical positions produce major climatic disparities across different regions of the world and generate different drought distributions throughout the country. Fig. 3 represents the spatial distribution of climate stations over China.

4.2. Data description

To validate our methodology, different types of data were used.

Remote sensing data: The LST dataset was extracted from MOD11A2, 8 days with 1 km spatial resolution and were gathered from NASA-EOSDIS between 2000 and 2020 [18] the NDVI data were collected from MYD13A2 16-days with 1km spatial resolution from 2000 to 2020 [6] and TRMM precipitation data are collected from the daily product 3B42 of TRMM dataset with 0.25°x 0.25° spatial resolution[7].

Biophysical data: Soil moisture data are based on 0.25° x 0.25° monthly GLDAS-NOAH025-M.2.1 dataset from 2000 to 2020 [2, 13]

Climate data: SPEI is based on climate data. The SPEI is used to determine the duration of drought and allows comparison of drought severity through time and space [**?**]. The collected SPEI data have four different spatial resolutions (.i.e. 1, 3, 6, 12 months), the ETP data are daily gathered. This feature may be calculated using many formulas, in our case we used the Penman-

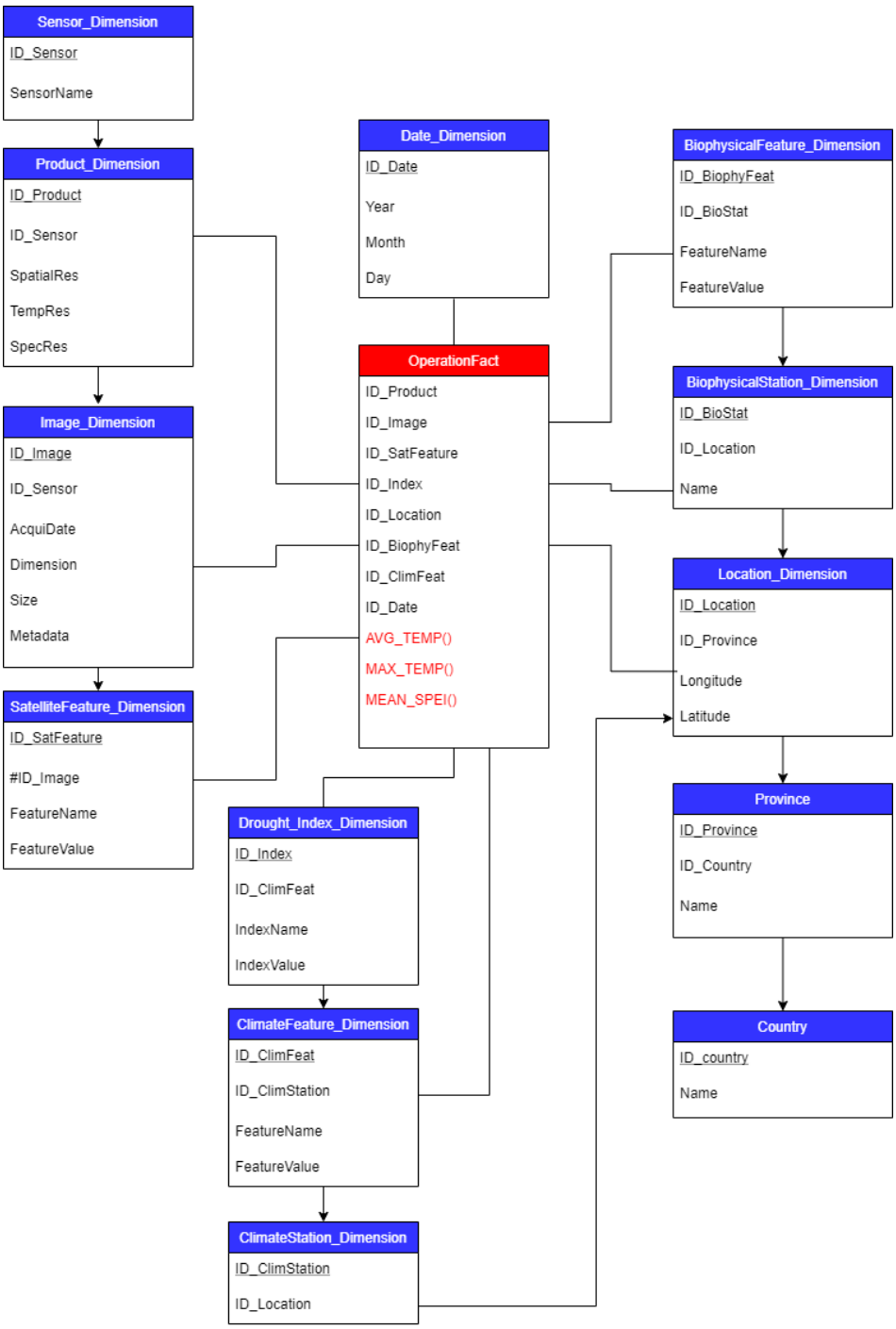

Fig. 2. Multidimensional conceptual schema for drought big data warehouse

Monteith formula, and finally, stations climate data (.i.e. Temperature, Humidity, Pressure, Sunshine duration, Wind speed) were collected from 511 stations over China.

4.3. DW Implementation

Table 2 compares 3 different big data storage tools; Hive, MongoDB, and Cassandra basing on several technical features. Hive handles almost all the features needed for the construction of the proposed methodology. Consequently, Apache Hive was implemented for DW building in this

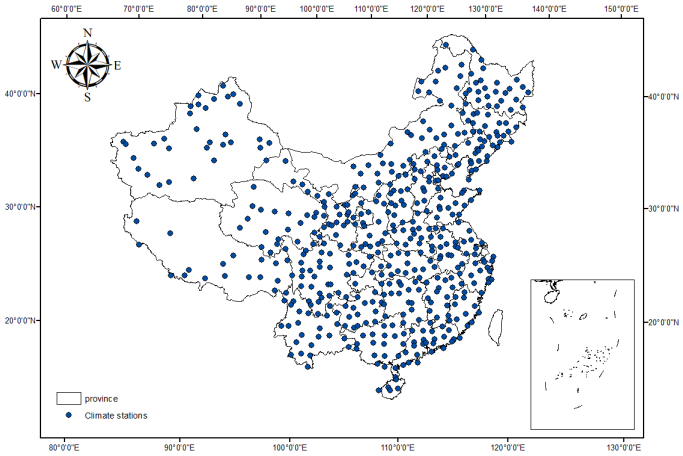

Fig. 3. Spatial distribution of climate station over China

paper. It is reported to several points. First of all, Hive is built with Apache Hadoop which is the most powerful cloud computing platform for Big Data. Besides, Hive supports high storage capacity, business intelligence, and data science. Also, Hive provides APIs for external data query management, thus, here we used HiveQL and ElasticSearch for data and query management. To test the proposed architecture some queries were illustrated. The main objective of

Table 2. Comparison between Hive, MongoDB and Cassandra

Features	**Hive**	**MongoDB**	**Cassandra**
Governance	Yes (using Hadoop)	Yes	Yes
Monitoring	Yes	Yes	Yes
Replication	Yes	Yes	Yes
Data Schema	Yes	No-schema	Flexible Schema
Ad-hoc	Yes	Yes	No
Data lifecycle management	Yes (Via hadoop)	Yes	Yes
ETL	Yes	No	Limited
Structure	Master-slave	Master-slave	Peer-to-Peer
High Storage capacity	Yes	Yes	Yes
Data Ingestion	Yes	Yes	No
Business Intelligence	Very Good	Limited	Good
High performance	Non-real time	Real-time	Real-Time

these queries is to provide information about drought duration and drought severity. Table 3 represents an example of queries. Several tests were performed on an Ubuntu 16.04.4 LTS x64-based computer with Intel(R) Core(TM) i7-7700HQ processor, 2.80GHz CPU, 16 GB of RAM with Java version 1.8. Data loading was performed using Apache Flume 1.9 and Apache Hive 2.3 built on Hadoop 2.7. Apache Flume provides Hive Sink for Hive tables management. These sink streams events containing data directly into a Hive partition.

4.4. Results and Interpretation

To perform the data loading module 3 different ways were tested. The first one is the default configuration of Apache Flume with three replicas, the second one is the customization of the data block size with 1GB for one block and the last one is our proposition 1GB in a parallel

Table 3. Queries examples

Query description	Queries examples
Fist Query example: The average of drought index SPEI-1 in 2000	SELECT AVG(ID.IndexValue) FROM OperationFact OF, Date_Dimension D, Index_Dimension ID WHERE D.ID_Date=OF.ID_Date and OF.ID_Index= ID.ID_Index and D.Year="2000" and ID.IndexName="SPEI-1";
Second Query example: The annual precipitation average between 2000 and 2019. For the humidity and temperature.The CFD.FeatureName must be changed to the right feature name.	SELECT AVG(CFD.FeatureValue) FROM OperationFact OF,Date_Dimension D, Climate_Feature_Dimension CFD WHERE D.ID_Date=OF.ID_Date and OF.ID_Index= CFD.ID_Index and CFD.FeatureName="Precipitation" GROUP BY D.Year;
Third Query example: The province having the minimum SPEI-3 value in 2019	SELECT PR.name FROM Province PR, Location_Dimension LD, Drought_Index_Dimension DID, OperationFact OF, Date D WHERE PR.ID_Province=LD.ID_Province and LD.ID_Location=OF.ID_Location and DID.ID_Index=OF.ID_Index and OF.ID_Data=D.ID_Date and D.Year=2019 and DID.IndexValue=(SELECT MIN (IndexValue) FROM Drought_Index_Dimension WHERE IndexName="SPEI-3");
Fourth Query Example: Provinces having NDVI value less than 0 in ascending order	SELECT PR.ID_PROVINCE, PR.Name FROM SatelliteFeature_Dimension SFD, Province PR, OperationFact OF, Location_Dimension LD WHERE OF.ID_SatFeature=SFD.ID_SatFeature AND PR.ID_Province=LD.ID_Province AND LD.ID_Location=OF.ID_Location AND SFD.FeatureName='NDVI' Group By PR.Name Having SFD.FeatureValue <0 ORDER BY PR.Name ;

way. Fig. 4 shows the results which revealed that the proposed architecture gave the best performance. The time consumed in the proposed architecture is less than the time consumed for default configuration with x2.5 and x4 for the manual block size configuration.

Table 3 illustrates different queries used for testing our proposed approach. In every query,

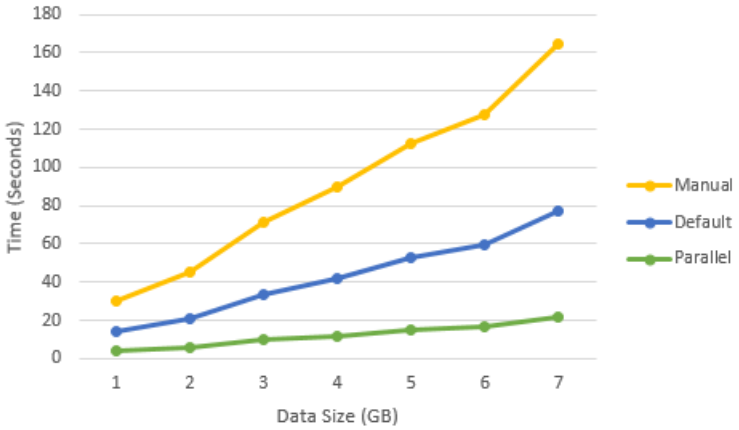

Fig. 4. Comparison between time consumed in the data loading module using the default Apache flume configuration, the manual configuration and the proposed parallel architecture.

several commands were used such as WHERE, GROUP BY, HAVING, LEFT (RIGHT) JOIN, ORDER BY, and UNION. The combination of these commands could affect the runtime of the queries. Table 4 represents the number of parameters used in every query. Fig. 5 illustrates the runtime in seconds of each query. The results revealed that when the number of the used

Table 4. Number of parameters used in each query

Queries	Number of commands	Number of tables
Q1	3	3
Q2	3	3
Q3	2	5
Q4	5	4

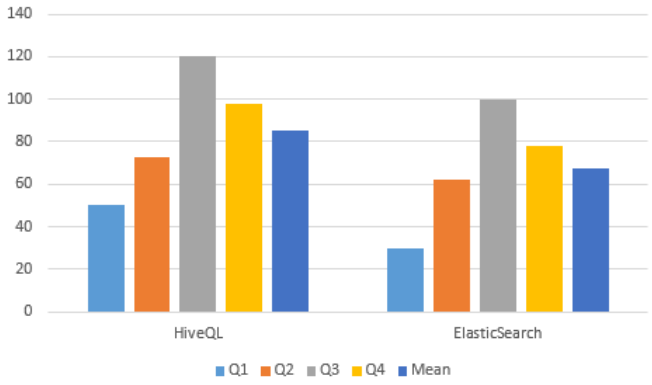

Fig. 5. Comparison between the queries runtime.

command increase, the runtime increase. Besides, the number of the used tables in a query affects the runtime speed. In fact, in Q3 and Q4, 5 and 4 tables and 2, 5 commands were used (respectively) but the results showed that the runtime of Q3 was higher than Q4.
These results are the same even when ElasticSearch was used for querying. Besides, the use of ElasticSearch for SQL queries gave better results than HiveQL in terms of time response. Thus, we conclude that both are suitable for the drought data analysis as they can handle all types of

data (i.e. textual and numerical data, geospatial data, and structured and unstructured data). For the visualization, several forms are used. Maps are used to show the Spatio-temporal variation of drought intensity over China. Fig. 6 illustrates drought mapping in China in 2000 and 2019.

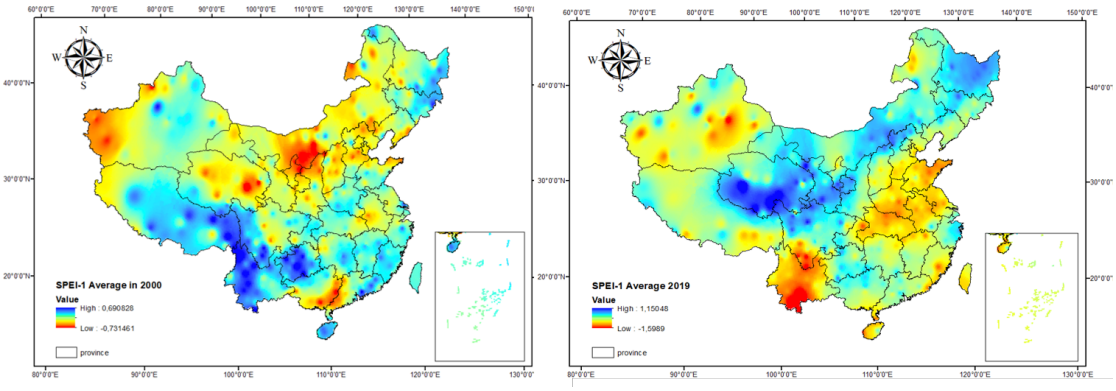

Fig. 6. Visualization of the SPEI-1 Average in 2000 and 2019

The results of the first application in 2000 and 2019 show that the average of SPEI-1 varies between 0.69 and -0.7 in 2000 and 1.5 and -1.5 in 2019. Fig. 6 shows that some provinces moved from WET (regions in dark blue) to dry (regions in red) basing on the SPEI values. For example, Southwest China is in a wetter condition in 2000 but in 2019 and Northeast China is in a drier condition in 2000 and a wetter condition in 2019.

5. Conclusion

Due to the increase of environmental disasters, efficient use of Earth observation data is needed for monitoring, forecasting, and prevention. The volume of these data continues increasing, thereby, scientists and decision-makers faced many challenges in collecting, pre-processing, storing, and processing this huge amount of heterogeneous data. These data are used for disaster management. In this paper, we propose an architecture for spatio-temporal data management. The architecture is composed of four modules: data collection, data preprocessing, data storage and interpretation, and decision making. The proposed architecture is based on spatio-temporal big data warehouse for drought data management. To load the data into the Hadoop systems, Apache Flume was adopted in a parallel way to accelerate the data ingestion and improve the efficiency of the overall system. A snowflake schema was adopted for the integration of spatio-temporal data in the data warehouse. The schema is scalable and compatible with Big Data modeling for other natural hazard prevention. Based on this scheme, we extracted, migrated, and loaded information from different datasets into a single representation of the drought dataset. This dataset was requested using various SQL-like queries through HiveQL and ElasticSearch. The parallel loading with a customized block size revealed better performance than the manual loading and the default loading (non-parallel) with almost x4 and x2.5 (respectively) improvement in the time consumed during the ingestion module. For future works, we aim to develop a new web service-based tool to provide real-time information in Spatio-temporal scales, with a user interface that could help the decision-makers to interrogate and extract different knowledge from the proposed system.

ACKNOWLEDGEMENT

The authors would like to thank Yanxin Zhu for providing the climate data and shape files.

References

1. Balti, H., Ben Abbes, A., Mellouli, N., Farah, I. R., Sang, Y., & Lamolle, M. (2020). A review of drought monitoring with big data: Issues, methods, challenges and research directions. Ecological Informatics, 60. https://doi.org/10.1016/j.ecoinf.2020.101136
2. Beaudoing, H., & Rodell, M. (2016). GLDAS Noah Land Surface Model L4 Monthly 0.25 x 0.25 degree V2.1. Nasa/Gsfc/Hsl, 92(November), 607–615.
3. Chang, K. (2019). Geographic Information System. In International Encyclopedia of Geography (pp. 1–10).
4. Corr, L., & Stagnitto, J. (2011). Agile Data Warehouse Design: Collaborative Dimensional Modeling, from Whiteboard to Star Schema. 330.
5. Dageville, B., Cruanes, T., Zukowski, M., Antonov, V., Avanes, A., Bock, J., ... Unterbrunner, P. (2016). The snowflake elastic data warehouse. Proceedings of the ACM SIGMOD International Conference on Management of Data, 26-June-20, 215–226.
6. Didan, K. (2015). MYD13A2 MODIS/Aqua Vegetation Indices 16-Day L3 Global 1km SIN Grid V006 [Data set]. NASA EOSDIS Land Processes DAAC.
7. Huffman, G. J., Pendergrass, A., & Staff, N. C. for A. R. (1998). The climate data guide: Trmm: Tropical rainfall measuring mission.
8. Kechadi, M. T., Le Khac, N. A., & Ngo, V. M. (2020). Data warehouse and decision support on integrated crop big data. International Journal of Business Process Integration and Management, 10(1), 17.
9. Liu, S., Han, C., Wang, S., & Luo, Q. (2012). Data warehouse design for earth observation satellites. Procedia Engineering, 29, 3876–3882.
10. Merritt, P., Bi, H., Davis, B., Windmill, C., & Xue, Y. (2018). Big Earth Data: a comprehensive analysis of visualization analytics issues. Big Earth Data, 2(4), 321–350.
11. Mishra, A. K., & Singh, V. P. (2011). Drought modeling - A review. Journal of Hydrology, 403(1–2), 157–175.
12. Ngo, V. M., & Kechadi, M. T. (2020). Crop knowledge discovery based on agricultural big data integration. ACM International Conference Proceeding Series, 46–50.
13. Rodell, M., Houser, P. R., Jambor, U., Gottschalck, J., Mitchell, K., Meng, C. J., ... Toll, D. (2004). The Global Land Data Assimilation System. Bulletin of the American Meteorological Society, 85(3), 381–394.
14. Overpeck, J. T., Meehl, G. A., Bony, S., & Easterling, D. R. (2011). Climate data challenges in the 21st century. Science, 331(6018), 700–702.
15. Schuetz, C. G., Schausberger, S., & Schrefl, M. (2018). Building an active semantic data warehouse for precision dairy farming. Journal of Organizational Computing and Electronic Commerce, 28(2), 122–141.
16. Schulze, C., Spilke, J., & Lehner, W. (2007). Data modeling for Precision Dairy Farming within the competitive field of operational and analytical tasks. Computers and Electronics in Agriculture, 59(1–2), 39–55.
17. Solodovnikova, D., & Niedrite, L. (2018). Towards a Data Warehouse Architecture for Managing Big Data Evolution. DATA 2018 - Proceedings of the 7th International Conference on Data Science, Technology and Applications, 63–70.
18. Wan, Z., Hook, S., & Hulley, G. (2015). MOD11A2 MODIS/Terra Land Surface Temperature/Emissivity 8-Day L3 Global 1km SIN Grid V006 [Data set]. NASA EOSDIS Land Processes DAAC.
19. Yao, X., Li, G., Xia, J., Ben, J., Cao, Q., Zhao, L., ... Zhu, D. (2020). Enabling the big earth observation data via cloud computing and DGGS: Opportunities and challenges. Remote Sensing, 12(1).