Conceptual Analysis of Big Data Using Ontologies and EER

Kulsawasd Jitkajornwanich[1] and Ramez Elmasri[2]

[1]Geo-Informatics and Space Technology Development Agency (Public Organization)
120 The Government Complex, Chaeng Wattana Rd, Lak Si
Bangkok 10210, Thailand
kulsawasdj@gistda.or.th

[2]Department of Computer Science and Engineering
The University of Texas at Arlington
701 S Nedderman Dr
Arlington, TX 76019, USA
elmasri@cse.uta.edu

Abstract. Large amounts of "big data" are generated every day, many in a "raw" format that is difficult to analyze and mine. This data contains potential hidden meaningful concepts, but much of the data is superfluous and not of interest to the domain experts. Thus, dealing with big raw data solely by applying a set of distributed computing technologies (e.g., MapReduce, BSP [Bulk Synchronous Parallel], and Spark) and/or distributed storage systems, namely NoSQL, is generally not sufficient. Extracting the full knowledge that is hidden in the raw data is necessary to efficiently enable analysis and mining. The data needs to be processed to remove the superfluous parts and generate the meaningful domain-specific concepts. In this paper, we propose a framework that incorporates conceptual modeling and EER principle to effectively extract conceptual knowledge from the raw data so that mining and analysis can be applied to the extracted conceptual data.

Keywords: conceptual modeling; big data; NoSQL; distributed computing

1 Introduction

Enormous amounts of data are rapidly generated every day in almost every application domain. These raw data could be thought of as a huge data warehouse, which contains hidden and meaningful information. However, to analyze the available raw data directly from its original formats is not easy as the data is often in a format that is difficult to analyze and is usually 'big'.

Although there is no formal definition of big data, 3V definition: volume, variety, and velocity, is often used to describe big data by its characteristics [1][28]. In any given domain, big data contains potential hidden meaningful concepts as well as superfluous data that are not of interest to the domain experts. As a result, dealing with

big data solely by applying a set of distributed computing technologies such as MapReduce [2], BSP (Bulk Synchronous Parallel) [3], and Spark [4], and/or distributed storage systems namely NoSQL databases [7] may not be an efficient way to discover the knowledge hidden in the data. To enable analysis, the big data need to be pre-processed so that the superfluous parts are removed (also known as a "cleaning" of the raw data) and the meaningful domain-specific knowledge is extracted.

Ontology, a specification of conceptualization [14], has practically been used in knowledge modeling as it allows domain-specific knowledge to be formalized and reasoned about in a logical way. ER (Entity-Relationship) and EER (Enhanced-ER) models/diagrams are excellent tools to communicate concepts, and can also be easily converted to relational tables. Our goal is to enable big data in any given domain to be analyzed by utilizing conceptual modeling (through ontologies) and EER to represent the domain-specific knowledge in the data. The formalized concepts are developed based on consulting with domain experts in the area of knowledge covered by the raw data.

An overview of our framework is shown in Fig. 1. The advantages of our framework include the capture of domain-specific conceptual knowledge (which is typically much smaller in size, compared to the raw data), accommodating existing traditional analysis as well as facilitating new knowledge discovery, and better system performance by applying distributed technologies (e.g., map-reduce, HDFS [Hadoop Distributed File System], etc.) to clean and convert the raw data. Our framework also offers more robust and user-friendly analysis by storing the final conceptual knowledge in a relational database.

The organization of this paper is as follows. We describe our framework in Section 2. A case study that adopted the framework is discussed in Section 3. We use big raw rainfall data in our case study. Finally, conclusion and future work are discussed in Section 4. In the case study, we briefly describe formalization of rainstorm ontology concepts [10][11][12] and translate them into EER. The mapping algorithms are implemented and the comparison experiments are performed. We also give some examples of how analysis and mining can be done on the resulting conceptual storm data.

2 Framework Description

There are four main components in our framework as shown in Fig. 1, each of which is described in the following subsections.

2.1 Developing and Formalizing Domain-specific Concepts into an Ontology with the Assistance of Domain Experts

The first process is to study a particular domain (where the big raw data comes from), come up with the domain-specific concepts, and formalize them into an ontology. This requires a literature review in the application domain as well as working with the domain experts to determine the important concepts that are needed. Investi-

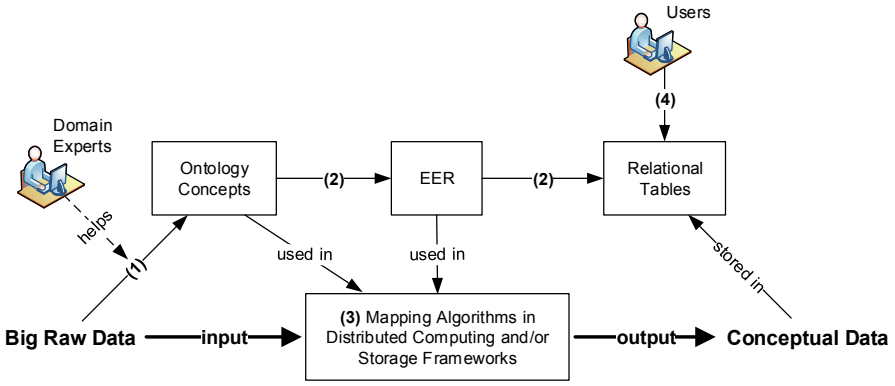

Fig. 1. Framework architecture

gation as to how their research is currently conducted using the traditional data processing methods is also required; some traditional methods (or more complex/previously-impossible analysis) could be improved (or realized) by utilizing available big data analysis tools such as map-reduce/hadoop® [6].

The developed ontology must satisfy the domain experts' requirements and capture not only the essential concepts that they are looking for but also other potential concepts–which may not have been previously identified, but could be of benefit to them. This can also help the domain users to better understand their own datasets. The hidden insight and conceptual relationships can consequently lead them to the knowledge that was not previously discovered. Thus, further complex analysis and mining can be applied.

2.2 Translating the Domain-specific Ontology to EER and Mapping the EER to Relational Tables

In the context of big data, an RDBMS (Relational Database Management System) is usually not a preferred option and often labeled as incompatible with the needs of big data analysis and mining. Their key features, however, are well recognized, in term of user-friendly analysis capabilities. The concepts of NoSQL databases, on the other hand, spread rapidly and caught a lot of attention as tools for big data storage and analysis/mining in the past few years (as of now, there are more than 100 different NoSQL databases available in the market [7]). The main advantages of NoSQL databases include high availability, fast key-value access, horizontal scalability, fault-tolerance, and dynamic/semi-structured data type support.

However, the emerging of NoSQL does not primarily intend to replace the conventional relational database but instead complement it as one application may work for one system but not for the other depending upon the application scenario. Some features of RDBMS are not readily available on NoSQL such as strong consistency, full

support relational features (e.g., join, group by) "across" partitions/documents, normalization, and fully-developed declarative query language.

Our framework combines the benefits from both systems (RDBMS and NoSQL) together. We use a distributed storage system (HDFS) during the processing steps and load the final conceptual outputs into a relational database.

To store the final conceptual outputs in a relational database, we translate the formalized domain concepts from the previous process to an EER model, which will later be mapped to relational tables. The general ideas of ontology-to-EER translation are as follows:

- An object in ontology becomes an entity in EER, and a class becomes an entity type.
- A datatype property in ontology becomes an attribute or derived attribute in EER.
- An object property in ontology becomes an attribute that is generally a foreign key. However, in some cases depending on the application ontology, an object property can be mapped to an entire new entity to better suit the required types of analysis (e.g., storm centers and storm tracks as seen in Section 3.2: Fig. 4 and 5).
- Other ontology concept specifications are translated through the use of EER features such as primary and foreign keys, cardinality and participation constraints, stored procedures, UDFs, triggers, etc.

The EER-to-relational tables mapping process is done by using the techniques described in [14].

2.3 Designing and Implementing Mapping Algorithms to Convert the Raw Data to the Conceptual Data

To design mapping algorithms, four main factors are taken into account: 1) structure and format of the big raw data, 2) choices of distributed computing/storage framework, 3) domain-specific ontology, and 4) EER model corresponding to the ontology.

Understanding the structure and format of the big raw data helps in optimizing the computation, I/O, and buffer usage in the raw data-to-conceptual data mapping algorithms. Three aspects of big raw data are considered: data representation (i.e., examining how the raw data is formatted and interpreted), data transmission (i.e., determining delivery method, transmission frequency, and downtime period of the raw data), and data integrity (i.e., ensuring the consistency of the raw data). Next, we make the decision as to which distributed technology should be used. The selected technology should take full advantage of the characteristics of the raw data as well as other available resources (e.g., hardware). The ontology is used to ensure the formalized domain concepts are correctly identified. Finally, the corresponding EER model is used to convert the final conceptual outputs into relational database-compatible format. In addition, since the final conceptual data is now stored in a relational database, the verification process can also be done through SQL.

2.4 Performing Analysis and Mining on the Conceptual Relational Data

After the algorithms are executed on the big raw datasets, we now have the extracted conceptual data stored in a relational database. The size of the conceptual relational outputs are usually significantly reduced when compared to the size of the big raw data as the superfluous parts are removed and the raw data is summarized/converted into meaningful domain-specific concepts. The analysis and mining tasks can then be easily conducted by a domain user on the conceptual knowledge base both directly via SQL and indirectly by extracting the conceptual data from the relational database.

3 Case Study: Rainfall Precipitation Data

In this section, we show a case study to which the proposed framework has been adopted. Our big raw data is rainfall precipitation datasets, *MPE (Multisensors Precipitation Estimate)*, retrieved from NOAA: National Weather Service (NWS) in the hydrology domain. We describe how each process of the framework is applied to our big raw rainfall data in the following subsections.

3.1 Rainstorm Formalization

In the hydrology domain, rainfall precipitation data is one of the hydrological observation types that most hydrologists work with (they also work with other types of datasets, such as soil moisture, river/stream levels, and watersheds). Rainfall-related analysis usually involves three characteristics of the data: rainfall statistical properties, correlation among storm characteristics (such as DDF: Depth-Duration-Frequency [13]), or focusing on the extreme precipitation values. However, the majority of these tasks are based on a location-specific analysis [13][15][16]. This prevents the analysis tasks from extracting storm-specific (or called 'overall' [10][11][12]) spatio-temporal aspects of the storms, which include storm movement (trajectory) and speed. After consulting with the hydrology experts, we introduced a partial rainstorm ontology that can eliminate this limitation and also support traditional location-based analysis. In the resulting ontology, three formalizations of rainstorms are introduced [10]: *local storm*, *hourly storm*, and *overall storm*.

Our ontology formalization considers the raw data framework, which we briefly describe now. The rainfall precipitation data is recorded hourly in a text file format. The data is reported for a regular grid structure called HRAP: Hydrologic Rainfall Analysis Project [17][18], where each grid location is approximately 4 km by 4 km (see Fig. 2). Each grid (site) location has a unique identifier, from which the lat/long coordinates can be derived. The format of the raw data contains four attributes: row number in the text file, site location, precipitation value, and observation time. Each line in the data file represents a precipitation value at a particular site during a particular hour.

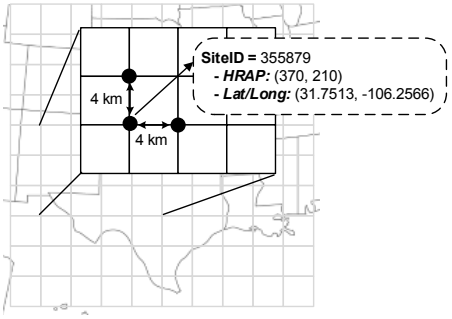

Fig. 2. Example of site location in HRAP

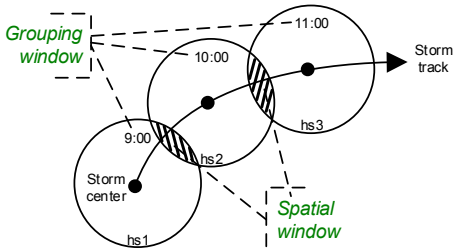

Fig. 3. An overall storm and its hourly storms

Informally, a *local storm* is a site-specific storm, which considers each site location independently when analyzing a storm. An example of local storms is the sequence of storms (separated by a certain number of hours with zero precipitation—inter-event time [15][19][20]) that occurred at site location 586987 last month. An *hourly storm* has an orthogonal concept to local storm. It considers a specific time point (hour) instead of a particular site location, and includes all contiguous locations that have precipitation during the same hour. The last storm formalization, *overall storm*, considers both location and time together when analyzing a storm. The result is the capture of storm as a whole, allowing storm movement, speed, and other overall storm characteristics that could not be found in most hydrology papers to be materialized [13][15][16][19][20]. An overall storm can be viewed as a sequence of hourly storms as they progress through time, as long as they have a spatial overlap from one hour to the next. The combination of hourly storms that form an overall storm satisfies two requirements: *grouping window* and *spatial window* [10]. Fig. 3 shows an example of how an overall storm moves over time.

Each of the three types of rainfall concepts will have specific features. We will discuss these in the next section, when we develop the EER diagrams for these concepts.

3.2 Ontology Translation and Mapping

To translate the domain ontology to an EER model for storing the final conceptual outputs in a relational database, we take into account the formalized domain-specific concepts and the characteristics of the raw rainfall data, and use the methodology described in [14]. Our rainstorm ontology is translated to an EER, which later be mapped to the database schema (as shown in Fig. 4, 5) containing 8 relational tables.

`LocalStormHours` stores local storms information for each hour of every site; its characteristics are summarized into `LocalStorms`. Each local storm is uniquely identified by (`YearID`, `LSID`); `YearID` indicates a particular year where the local storms are identified. `HourlyStormSites` and `HourlyStorms` contain precipitation value for each site of an hourly storm, and its statistics, respectively. Each hourly storm can be uniquely identified by (`DatetimeUTC`, `HSID`) as the rainfall data files are recorded hourly and each file is independent from others.

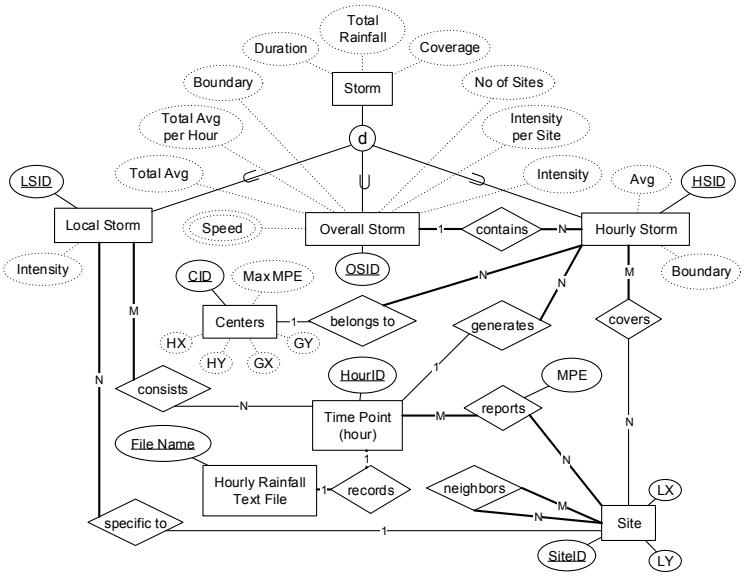

Fig. 4. EER diagram for rainstorm ontology

`OverallStormHourlyStorms` stores information of all hourly storms combined into an overall storm. The primary key (denoted by underline "__") for this table is (`DateTimeUTC`, `HSID`) because an hourly storm can belong to only one overall storm. Technically, this table could be further mapped to `HourlyStorms` by having (`YearID`, `OSID`) as a foreign key (denoted by arrow "→") to `OverallStorms`. However, we mapped it into a separate table because the hourly storm statistics (`HourlyStorms`) is calculated at the end. Having a separated `OverallStormHourlyStorms`, we do not need to wait until hourly storm identification is concluded in order to identify overall storms. Hourly storm and overall storm identifications can run concurrently. `OverallStormTracks` contains track information of the overall storms for each hour. Note that the `OverallStormTracks` table is derived from the relationships among the entity types: `Overall Storm`, `Hourly Storm`, `Centers`, and `Time Point` in Fig. 4.

3.3 Implementation of Storm Identification System

In this process, we developed mapping algorithms designed specifically to take full advantage of the structure and format of the big raw rainfall data in extracting its domain-specific ontology. The map-reduce (MR) paradigm is selected as our distributed computing framework as it is one of the well-known and efficient tools in analyzing big data across multiple machines. Our implemented mapping algorithms are called *MR-based Storm Identification System* [11][12]. Three main components of the system consist of MR-based- local storm identification (MR-LSI), hourly storm identification (MR-HSI), and overall storm identification (MR-OSI). With map-reduce, we can efficiently utilize our cluster of 19 servers as we can see in our comparison experiment between previous (non-MR) approach and MR-based approach (see Table 1).

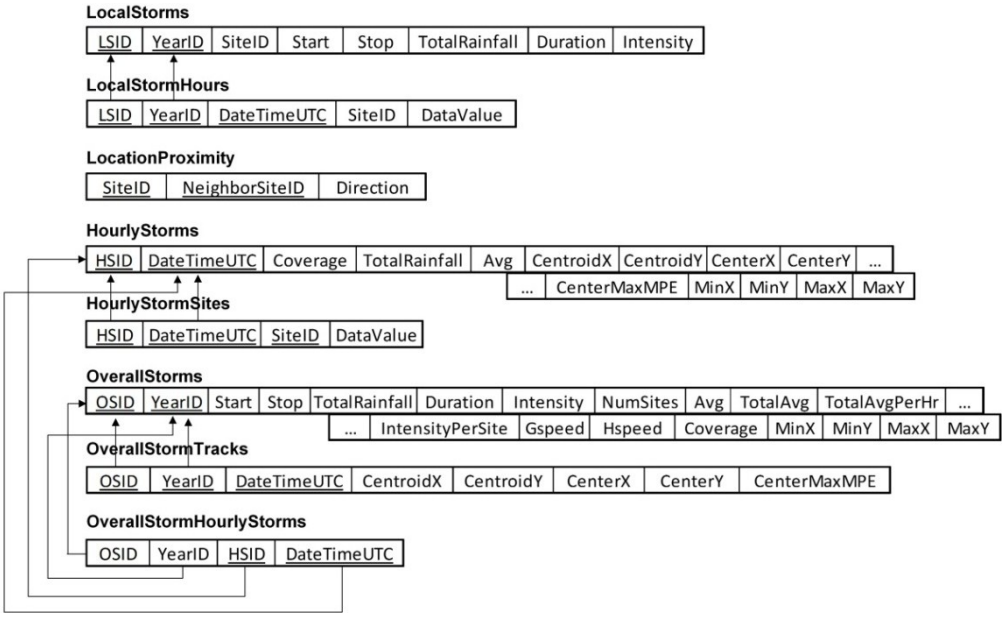

Fig. 5. Database schema corresponding to the rainstorm EER

To compare the system performance of the MR-based approach [11][12] with the previous (non-MR) approach, we use a rainfall dataset (MPE) from October 2010 to December 2011 covering 37,413 site locations in Texas, retrieved from NOAA-NWS [10]. In the non-MR approach, the dataset is resided in a relational database and the identification process is done on a single server. The server runs on Microsoft® Windows Server® 2008 Enterprise OS with 2.83 GHz Intel® Xeon® quad-core processors, 20 GB of RAM, 500 GB of local disk, and 10 TB of external disk. In the MR-based approach, we use the same dataset but is in a textual format (which covers much more site locations [i.e., Texas and some surrounding areas]). The experiment is done on a Hadoop® cluster of 1 master node and 18 worker nodes. All nodes have the same hardware specification (except local disks: 1.5 TB in worker node and 3 TB in master node): 3.2 GHz Intel® Xeon® quad-core processors and 4 GB of RAM. The cluster is operated by ROCKS Cluster 6.3 OS and has Hadoop® 1.0.3 installed in every node.

The computational time comparison between the non-MR implementations and the MR-based implementations is shown in Table 1. The experiments of the two approaches give the same results but the MR-based approach is executed significantly faster. The MR-based approach allows programs to be executed distributedly on multiple machines and hence the efficiency of the storm analysis is increased.

Table 1. Comparison of processing time between non-MR and MR-based approaches

Regions / Number of Raw Data Records	Number of Sites	Processing Time (in hrs.)					
		Previous Non-MR Approach			MR-based Approach		
		LSI	HSI	OSI	LSI	HSI	OSI
1. East Texas (48,953,130)	4,643	8.67	1.44	1.24	<---- less than 3 hrs. for all sites (ind. 37,413 sites in Texas) ---->	<---------- less than an hour for all sites ---------->	<---------- less than 3 hrs. for all sites ---------->
2. Edwards Plateau (73,415,532)	6,962	8.72	1.23	2.25			
3. High Plains (31,711,927)	3,008	4.5	0.32	0.57			
4. Low Rolling Plains (24,965,521)	2,368	3.35	0.28	0.27			
5. North Central (59,082,957)	5,604	8.66	1.17	1.64			
6. South Central (31,102,334)	2,949	4.28	0.67	0.42			
7. South Texas (31,949,386)	2,933	3.97	0.48	0.60			
8. Lower Valley (5,324,898)	601	0.55	0.07	0.08			
9. Trans-Pecos (65,136,216)	6,177	6.86	0.55	1.16			
10. Upper Coast (22,863,789)	2,168	3.88	0.57	0.39			
TOTAL	**37,413**	**53.44**	**6.78**	**8.62**			

3.4 Conceptual Analysis of Rainfall Data

In this section, we show some examples of how rainfall data can be analyzed by using our extracted storm data. From the experiment mentioned in Section 3.3, we extend it to the entire rainfall dataset for 16 years from 1997 to 2012. The conceptual storm data is extracted, summarized, and stored in a relational database. The size of the conceptual storm data is less than 1% of the size of the raw rainfall data as shown in Table 2.

We divide the analysis and mining tasks into two groups: 1) traditional hydrology analysis and 2) more flexible/robust analysis and mining [27]. We only discuss the first of these in this paper due to space limitations.

Most traditional rainfall analysis is based on location, meaning each site or region (set of sites) is considered separately when analyzing a storm. The goal is to investigate characteristics of storms at a particular location. These analyzed characteristics will then be used in creating an efficient/cost-effective hydraulic control structure (e.g., storm drain [to route localized runoff] and parking lot design for effective draining), and designing river flow or flooding prediction models [25]. The traditional rainfall analysis can be divided into three categories [13][15][16][25]: 1) storm statistical properties, 2) relationships between/among characteristics of storms, and 3) focusing on extreme precipitation values of storms. Some examples of the first item are as follows (for more examples in other analyses, please refer to [26]).

Table 2. Number of (conceptual) storm records in each component and year

Year	Num Raw MPE Records	MR-LSI Num Storms	MR-HSI Num Storms	MR-OSI Num Storms	Num Storm Records	Reduction in Raw Data Size (%)
1997	731,786,250	3,944,877	298,561	150,886	4,394,324	0.60
1998	1,451,804,250	6,372,104	455,575	235,409	7,063,088	0.49
1999	1,450,644,000	5,842,579	434,440	218,516	6,495,535	0.45
2000	1,453,627,500	6,138,978	439,557	225,029	6,803,564	0.47
2001	1,451,804,250	6,663,672	496,213	248,550	7,408,435	0.51
2002	1,451,804,250	6,827,462	448,670	196,531	7,472,663	0.51
2003	1,451,804,250	7,606,046	441,303	196,330	8,243,679	0.57
2004	1,455,782,250	12,526,769	545,125	237,457	13,309,351	0.91
2005	1,451,804,250	10,169,983	479,560	210,777	10,860,320	0.75
2006	1,450,478,250	10,354,175	519,978	227,517	11,101,670	0.77
2007	1,448,489,250	12,819,729	643,383	282,021	13,745,133	0.95
2008	1,455,782,250	10,371,608	544,036	248,001	11,163,645	0.77
2009	1,451,638,500	10,958,887	547,164	248,294	11,754,345	0.81
2010	1,451,141,250	10,108,909	546,926	247,449	10,903,284	0.75
2011	1,451,804,250	7,143,676	382,009	183,112	7,708,797	0.53
2012	1,455,782,250	9,024,720	481,920	225,075	9,731,715	0.67
TOTAL	**22,515,977,250**	**136,874,174**	**7,704,420**	**3,580,954**	**148,159,548**	**0.66**
AVERAGE	**1,407,248,578**	**8,554,636**	**481,526**	**223,810**	**9,259,972**	**0.66**

In storm statistical properties analysis, each characteristic of storms is analyzed separately for its statistical properties. The storm characteristics include inter-event time, total rainfall, and duration. There are six main statistics studied: mean (average) inter-event time between storms, mean total rainfall at a particular location, number of storms during the study period, total duration of all local storms, distribution of total rainfall values over the various storms, and distribution of storm durations [15]. Each statistical property is analyzed for a particular inter-event time. Most statistics can directly be calculated using pre-computed attributes (such as query SQL1). For statistics that were not pre-computed, they can also be easily calculated by using SQL (see SQL2). In our analysis, we use $h = 6$ hours as the inter-event time. To determine the statistical properties for other inter-event times ($h = 8, 12, 16, \ldots$), we just need to change the inter-event-count parameter in the local storm identification program and re-run it.

SQL1. Determine mean total rainfall, number of storms, total duration for a given location

```
1:  SELECT    (AVG(TotalRainfall)|COUNT(*)|SUM(Duration))
2:  FROM      LocalStorms
3:  [WHERE    (SiteID = <site>|SiteID IN <region>)]
```

SQL2. Determine mean storm inter-event time for a given location (a site or region)

```
1:  SELECT    AVG(L2.Start–L1.Stop)
2:  FROM      LocalStorms L1 JOIN LocalStorms L2
3:            ON L1.YearID = L2.YearID AND L1.LSID = L2.LSID-1
4:  [WHERE    (SiteID = <site>|SiteID IN <region>)]
```

4 Conclusion and Future Work

In this paper, we propose a generalized analysis framework for big data in a given domain by utilizing conceptual modeling and EER. We show how the framework can be applied through a case study by using big raw rainfall datasets in the hydrology domain. In the rainfall case study, we overviewed the formalizations of the partial rainstorm ontology and translated it into an EER. The MR-based storm identification system is implemented based on the developed rainstorm ontology. Finally, we show some examples of how traditional hydrology analysis can be done by utilizing SQL capabilities over the conceptual storm data. In addition to traditional hydrology analysis, we are working on more complex mining tasks [27], such as pattern matching of storms based on storm shape, trajectory and directional analysis of storm progression using Markov models, and other approaches to identify and classify storms based on their characteristics. These results will be communicated in future work.

Although our ontology-guided framework can accommodate a domain-specific and user-friendly analysis of big data by converting raw data to conceptual data, the extracted conceptual data may be too large to fit in a relational database, even if it is distributed or parallel. Thus, there are options to either perform an analysis in a finer sub-domain or convert the EER/relations tables' concepts to a NoSQL system; in the latter option, some RDBMS features might be sacrificed.

References

1. Embley, D.W., Liddle, S.W.: Big Data—Conceptual Modeling to the Rescue. In: 32nd International Conference on Conceptual Modeling (2013)
2. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. In: 6th Symposium on Operating Systems Design and Implementation (2004)
3. Valiant, L.G.: A Bridging Model for Multi-Core Computing. In: 16th Annual European Symposium (2008)
4. Apache. Apache Spark™, http://spark.apache.org
5. Zou, B., Ma, X., Kemme, B., Newton, G., Precup, D.: Data Mining Using Relational Database Management Systems. In: 10th Pacific-Asia Conference (2006)
6. Lam, C.: Hadoop in Action. Dreamtech Press (2011)
7. nosql-database.org. List of NoSQL Databases, http://nosql-database.org
8. Amazon. Amazon DynamoDB, http://aws.amazon.com/dynamodb
9. MongoDB, http://www.mongodb.org
10. Jitkajornwanich, K., Elmasri, R., Li, C., McEnery, J.: Extracting Storm-Centric Characteristics from Raw Rainfall Data for Storm Analysis and Mining. In: 1st ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data (2012)
11. Jitkajornwanich, K., Gupta, U., Elmasri, R., Fegaras, L., McEnery, J.: Using MapReduce to Speed Up Storm Identification from Big Raw Rainfall Data. In: 4th International Conference on Cloud Computing, GRIDs, and Virtualization (2013)
12. Jitkajornwanich, K., Gupta, U., Shanmuganathan, S.K., Elmasri, R., Fegaras, L., McEnery, J.: Complete Storm Identification Algorithms from Big Raw Rainfall Data. In: 2013 IEEE International Conference on Big Data (2013)

13. Overeem, A., Buishand, A., Holleman, I.: Rainfall Depth-Duration-Frequency Curves and Their Uncertainties. Journal of Hydrology (2008)
14. Elmasri, R., Navathe, S.: Fundamentals of Database Systems, 6th ed. Pearson Education, (2010)
15. Asquith, W.H., Roussel, M.C., Cleveland, T.G., Fang, X., Thompson, D.B.: Statistical Characteristics of Storm Interevent Time, Depth, and Duration for Eastern New Mexico, Oklahoma, and Texas. Professional Paper 1725, U.S. Geological Survey (2006)
16. Lanning-Rush, J., Asquith, W.H., Slade, Jr., R.M.: Extreme Precipitation Depth for Texas, Excluding the Trans-Pecos Region. Water-Resources Investigations Report 98-4099, U.S. Geological Survey (1998)
17. NOAA's National Weather Service. The XMRG File Format and Sample Codes to Read XMRG Files, http://www.nws.noaa.gov/oh/hrl/dmip/2/xmrgformat.html
18. Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI). ODM Databases, http://his.cuahsi.org/odmdatabases.html
19. Asquith, W.H.: Depth-Duration Frequency of Precipitation for Texas. Water-Resources Investigations Report 98-4044, U.S. Geological Survey (1998)
20. Asquith, W.H.: Summary of Dimensionless Texas Hyetographs and Distribution of Storm Depth Developed for Texas Department of Transportation Research Project 0-4194. Report 0-4194-4, U.S. Geological Survey (2005)
21. National Oceanic and Atmospheric Administration (NOAA). National Weather Service River Forecast Center: West Gulf RFC (NWS-WGRFC), http://www.srh.noaa.gov/wgrfc
22. Unidata. What is the LDM?, https://www.unidata.ucar.edu/software/ldm/ldm-6.6.5/tutorial/whatis.html, 2014.
23. Chang, F., Dean, J., Ghemawat, S., Hsieh, W.C., Wallach, D.A., Burrows, M., Chandra, T., Fikes, A., Gruber, R.E.: Bigtable: A Distributed Storage System for Structured Data. In: 7th USENIX Symposium on Operating Systems Design and Implementation (2006)
24. NOAA. MPE: Multisensor Precipitation Estimate, http://www.erh.noaa.gov/marfc/Maps/xmrg/index_java.html
25. Mishra, S.K., Singh, V.P.: Soil Conservation Service Curve Number (SCS-CN) Methodology. Kluwer Academic Publishers (2003)
26. Jitkajornwanich, K.: Analysis and Modeling Techniques for Geo-Spatial and Spatio-Temporal Datasets. Doctoral Dissertation, The University of Texas at Arlington (2014)
27. Cheng, T., Haworth, J., Anbaroglu, B., Tanaksaranond, G., Wang, J.: Spatio-Temporal Data Mining. Handbook of Regional Science, Springer-Verlag Berline Heidelberg (2013)
28. IBM Big Data and Analytics Hub. Understanding Big Data: e-book, http://www.ibmbigdatahub.com/whitepaper/understanding-big-data-e-book.