Scientific Research Publishing

Journal of Computer and Communications, 2020, 8, 75-98
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Hybrid Warehouse Model and Solutions for Climate Data Analysis

Hasan Hashim

College of Computer Science and Engineering, Yanbu Taibah University, Taibah, KSA
Email: alhashimi2002@hotmail.com

How to cite this paper: Hashim, H. (2020) Hybrid Warehouse Model and Solutions for Climate Data Analysis. *Journal of Computer and Communications*, **8**, 75-98.
https://doi.org/10.4236/jcc.2020.810008

Received: September 29, 2020
Accepted: October 27, 2020
Published: October 30, 2020

Abstract

Recently, due to the rapid growth increment of data sensors, a massive volume of data is generated from different sources. The way of administering such data in a sense storing, managing, analyzing, and extracting insightful information from the massive volume of data is a challenging task. Big data analytics is becoming a vital research area in domains such as climate data analysis which demands fast access to data. Nowadays, an open-source platform namely MapReduce which is a distributed computing framework is widely used in many domains of big data analysis. In our work, we have developed a conceptual framework of data modeling essentially useful for the implementation of a hybrid data warehouse model to store the features of National Climatic Data Center (NCDC) climate data. The hybrid data warehouse model for climate big data enables for the identification of weather patterns that would be applicable in agricultural and other similar climate change-related studies that will play a major role in recommending actions to be taken by domain experts and make contingency plans over extreme cases of weather variability.

Keywords

Data Warehouse, Hadoop, NCDC Data Set, Weather

1. Introduction

Nowadays, the volume of data generated from different sources is rapidly increasing. Hence, administering and processing such a massive volume of data is challenging. It is time-consuming, costly, and has many obstacles to researchers involving it. The way of administering such data in a sense storing, managing, analyzing and extracting insightful information from the huge size of data is a challenging task. The volume of data produced on a daily basis is a massive amount

whereby this accelerating growth of data is due to the growth of the Internet of Things (IoT), Artificial Intelligence and Data Science. To make use of the huge volume of data, researchers in the domain often use cutting-edge techniques such as analytical tools and techniques with the help of Artificial Intelligence and machine learning methods. The main concern with big data is privacy, security, and discrimination. Researchers are then to start building support systems such as data warehouse.

The huge amount of weather data and climatic variables are recorded manually or digitally using many resources such as weather stations and satellites around the country. Thus, the storage and manipulation task of the information has to be effective, integrated and more flexible in Meteorological and climatology studies to achieve an effective and intensive analysis. The access to recorded Meteorological data varies according to the certain process, for instance, in weather forecasting tasks the raw data needed to be accessed rapidly and in Climatology, to increase the accuracy it is important to have high-quality historical weather data [1].

With technological advances in the area of tools and the different types of equipment to collect weather data, researchers can access and share huge data. Generally, the NCDC and Daily Global Weather Measurements 1901-2020 (GSOD, NCDC) is the largest active archive of climate data available online so far. In addition, the history of the NCDC dataset goes back to more than 150 years of data and the size of new data collected daily is close to 224 gigabytes. Moreover, it is easily accessible and downloads NCDC and GSOD datasets from NCDC website [2].

In the literature, it is suggested that a clear definition of a data warehouse which is a single, consistent, complete storage of data acquired from various sources in order to analyze it using a business intelligence tool [3] [4]. Data warehouse technology is applicable in a lot of domains in industry that use historical data for prediction, statistical analysis, and decision making, for instance, banking, consumer goods, finance industry, weather, healthcare, and Internet of Things (IoT) [1] [5] [6] [7] [8] [9]. Moreover, the huge amount of sensor data collected from the Meteorological domain has many problems to be addressed in the preprocessing stages. It demands careful manipulation so as to carry out weather analysis in an accurate way by extracting relevant patterns [10].

The traditional way of storing Meteorological data was file-based. Whereas, recently saving weather data in relational database management systems (RDMS) is attracting the attention of researchers in the area of climatology studies. As a result, many institutions operating in this domain are shifting towards storing their data in terms of RDMS [11]. Chen [12] describes a data warehouse framework namely Cheetah which is designed using the MapReduce platform. Dimri *et al.* [13] proposed a data warehouse model for storing weather data using On-Line Analysis Processing (OLAP) method to generate proper data for weather analysis and provide a multi-dimensional report. José *et al.* [14] developed a data warehouse to save climatic variables in the weather stations of Mexico. The au-

thors have used the SQL Server for storing the final version of data in the proposed model. Sameer and Madhu [15] discussed how to design a data warehousing model using Hadoop.

The main contributions of this work are:

- The main contribution of this work is the definition and implementation of hybrid data warehouse infrastructure to support the distribution of weather data storage, computing, and parallel programming.
- The new data warehouse can be implemented in different ways to store huge data sets and workloads for distribution in hybrid Hadoop platform.
- The main aim is to preserve and improve a traditional data warehouse for reporting, OLAP, and performance management while new development in data platforms for advanced analytics.
- The hybrid data warehouse model for climate big data enables for the identification of weather patterns that would be applicable in agricultural and other similar climate change-related studies that will play a major role in recommending actions to be taken by domain experts and make contingency plans over extreme cases of weather variability.

The rest of the paper is organized as follows. Section 2 presents a review of the related works in the data warehouse and big data context. Section 3 deals with the dataset description. Section 4 describes the concept and architecture of the data warehouse. Section 5 presents the concept of Big Data and the required tools such as Hadoop, Pig, Hive, and Sqoop. Section 6 shows the proposed data warehouse model for the weather data under consideration. Finally, concluding remarks and summaries are presented in Section 7.

2. Related Work

There are many research works that have been done to design a data warehouse based on Big Data and Hadoop framework.

Kalra and Steiner [7] explained the quality and content of data vary over time based on the types of information for instance, weather, and health data. Thus, this data needs to be gathered, processed and stored in different formats. The authors developed a weather data warehouse model that enables dynamic and smooth integration of new information sources and data formats. The proposed architecture depicts an active and flexible weather data warehouse that provides a broad variety of weather data from various sources to different weather-based applications.

Néstor *et al.* [16] discussed the problems of data recorded by meteorological which require strategies for capturing, delivering, storing and processing to increase the quality and stability of the data. They proposed a star schema model for a data warehouse that allows storage and analysis of historical multidimensional hydro-climatological data. Moreover, the proposed data warehouse provides efficient data storage in which data collected from two networks of hydro-meteorological stations goes back to more than 50 years in the city of Mani-

zales, Colombia.

Doreswamy *et al.*, [1] proposed a scalable architecture for a hybrid data warehouse approach for climate data using Hadoop and various big data tools. The proposed schema enables the identification of weather patterns and it is better to derive knowledge from the data in comparison to the traditional database.

Vuong *et al.* [17] show how designing and developing a data warehouse for the agriculture field has the main role in establishing a crop intelligence platform. They describe the requirements for efficient agricultural data-warehouses such as privacy, security, and real-time access among its stakeholders. Thus, the proposed system architecture and a database schema for designing and implementing an efficient agricultural data warehouse in Big Data and data mining.

3. NCDC Dataset Details

This section presents specific descriptions about the data produced by the NCDC dataset for Saudia country. The NCDC is a large dataset that has more than 9000 stations around the globe and is available online from NCDC meteorological site [2]. **Figure 1** shows the selected Saudi Arabia weather stations from the NCDC dataset and each station has 16 attributes. **Table 1** presents the column names and their corresponding description.

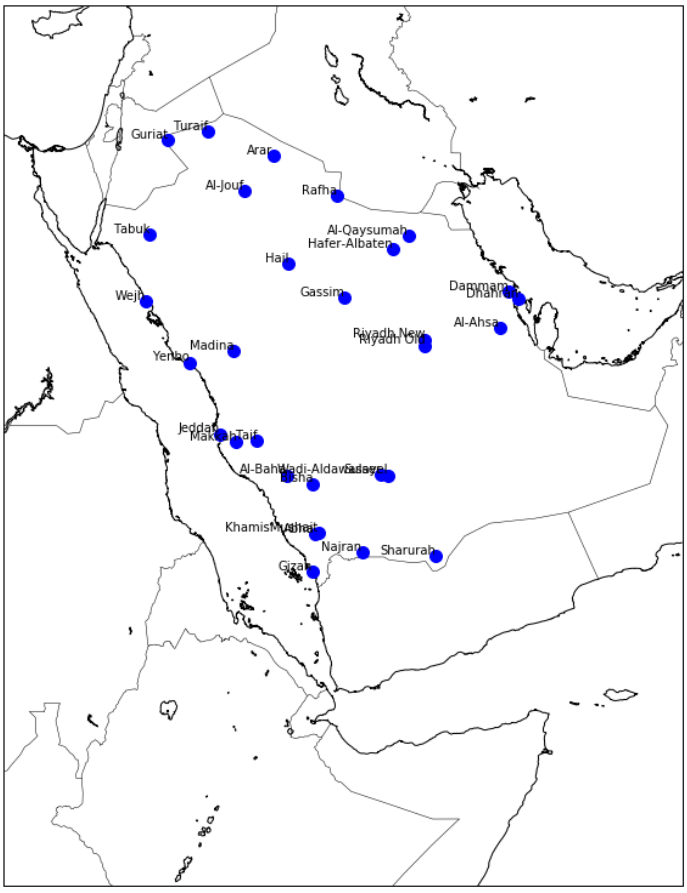

Figure 1. The selected stations of Saudi Arabia from NCDC dataset.

Table 1. Description of the attributes for each station.

No.	Attribute	Description
1	STN	Station number
2	WBAN	Weather Bureau Air force Navy
3	YEARMODA	The year, month and day
4	TEMP	Mean temperature
5	DEWP	Mean dew point
6	SLP	Mean sea level pressure in millibars
7	STP	Mean station pressure
8	VISIB	Visibility
9	WDSP	Mean wind speed in knots
10	MXSPD	Maximum sustained wind speed
11	GUST	Maximum wind gust
12	MAX	Maximum temperature
13	MIN	Minimum temperature
14	PRCP	Precipitation
15	SNDP	Snow depth in inches
16	FRSHTT	Fog rain snow hail thunder Tornado

Table 2 presents information such as station number, station name, latitude, longitude, begin and end date for an illustrative case of the selected Saudi Arabia weather stations from the NCDC dataset. Moreover, Table 3 demonstrates a sample data of one station of the NCDC dataset and the corresponding values for each feature. The problem associated with the NCDC dataset is that it has several missing values. The missing data for the selected attributes are taking the following values: 9999.9, 999.9 or 99.99. For example, a 9999.9 shows a missing value for the variables TEMP, DEWP, SLP, STP, MAX and MIN, whereas 999.9 indicates a missing value for the column names VISIB, WDSP, MXSPD, GUST and SNDP. Moreover, 99.99 shows a missing value for the column PRCP.

4. Data Warehouse (DW)

In the 1990s, Bill Inmon suggested the first architecture of data warehouse (DW) model. While Gartner, in 2005, provided a clear concept of the DW. The main task of DW schema is to accumulate and keep data from various sources for future analysis and decision making. Generally, the classic relational database schema is used to store, manage and query structured data. The present DW model is briefed as follows [4]:

- **Subject oriented**: In this case, the entire data is manipulated to classify into various domain areas, for instance, each domain will have complete data related to each subject.
- **Integrated:** This means that the following conditions should be satisfied: the logical model should be integrated and consistent. For instance, the values of the data should be standardized such as female/male representations should be consistent.

Table 2. The selected Saudi Arabia stations from the NCDC dataset.

STATION STN	STATION WBN	STATION Name	C	LATITUDE	LONGITUDE	ELEV	Begin DATE	End DATE
410060	99999	MUWAIH	SA	+22.433	+041.750	+0971.0	19851001	20110614
410080	99999	ZULM	SA	+22.717	+042.167	+0870.0	20041015	20041015
410200	99999	JEDDAH I.E.	SA	+21.417	+039.217	+0010.0	19910605	20050729
410240	99999	KING ABDULAZIZ INTL	SA	+21.680	+039.157	+0014.6	19830101	20161211
410260	99999	JEDDAH	SA	+21.500	+039.200	+0015.0	19560101	20061026
410300	99999	MAKKAH	SA	+21.433	+039.767	+0240.0	19830701	20161211
410360	99999	TAIF	SA	+21.483	+040.544	+1477.7	19830101	20161211
411140	99999	KING KHALED AB	SA	+18.297	+042.804	+2065.9	19830101	20161211
411280	99999	NEJRAN	SA	+17.611	+044.419	+1213.7	19830101	20161211
411400	99999	KING ABDULLAH BIN ABDULAZIZ	SA	+16.901	+042.586	+0006.1	19830101	20161211

Table 3. A sample of records and the attributes of one station.

STN-WBAN	YEARMODA	TEMP	DEWP	SLP	STP	VISIB	WDSP	MXSPD	GUST	MAX	MIN	PRCP	SNDP
410240 99999	20180101	75.1	59.1	1013.3	1011.4	6.2	12.0	20.0	999.9	78.8	69.8	0.00	999.9
410240 99999	20180102	73.1	47.5	1013.1	1011.1	6.2	8.4	12.0	999.9	78.8	66.2	0.00	999.9
410240 99999	20180103	77.0	49.4	1011.6	1009.7	6.0	7.2	15.0	999.9	87.8	67.6	0.00	999.9
410240 99999	20180104	74.7	58.2	1013.5	1011.5	6.0	6.1	15.0	999.9	80.6	69.8	0.00	999.9
410240 99999	20180105	71.4	53.3	1014.4	1012.5	6.2	7.6	14.0	999.9	78.8	66.2	0.00	999.9
410240 99999	20180106	71.5	52.8	1014.4	1012.5	6.2	9.9	15.0	999.9	80.6	65.5	0.00	999.9

- **Nonvolatile:** This means that the unmodified data shall be saved for long period of time in the DW.
- **Time variant:** In this case, the DW has the ability to store the newly modified versions of records.
- **Not virtual:** In this regard, the DW saves the data in the physical storage area for a longer period of time persistently.

The ETL stands for Extract, Transform and Load that stands for three database operations that are combined into one tool to extract data from different sources and place it into another data warehouse. The extract is the operation of reading data from multiple sources and different natures data such as structured, semistructured and unstructured data. The transform operation is used to fit the data into the analytical model. Moreover, the load operation deals with physically storing the data into the data warehouse. Finally, analysis, visualization and report generating are carried out as a final core activity in constructing a DW model (See **Figure 2**) [3].

4.1. The OLTP Data Processing

The Online Transactional Processing (OLTP) is a type of data processing technique

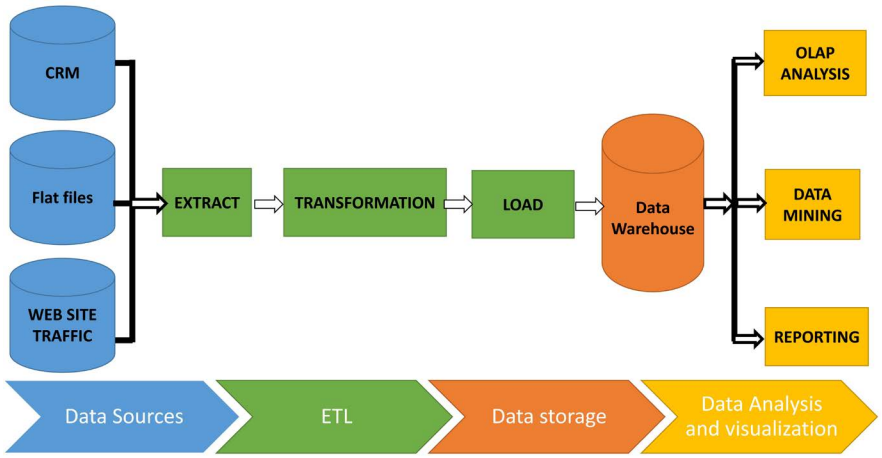

Figure 2. Traditional data warehouse [5].

that deals with transaction-related tasks and sub-tasks [18]. The common tasks in OLTP are inserting, updating, and deleting of data in a database by a large number of users concurrently. The OLTP handles recent operational data in small to medium-sized data with the goal to perform daily operations. It uses simple queries with read/write operations essential for faster transaction speeds. OLTP architecture is used to manage the day-to-day transaction of a business entity. The main aim of OLTP is for data processing and not for data analysis. Some example of OLTP related-system, such as the ATM center where it makes sure that withdrawal of a certain amount from ATM machine can not be more than the amount deposited in the saving account. Therefore, OLTP systems are highly optimized for transactional related tasks such as online banking, online flight ticket booking, sending a text message, order entry and adding a textbook to shopping carts [19].

The OLTP is known for the following two features namely concurrency and atomicity whereby the atomicity guarantees if a single step is incomplete during the transaction, then the entire process automatically stops. On the other hand, the concurrency property of the system prevents the altering of data by multiple users simultaneously. The main benefit of the OLTP system is it allows the administration of daily transactions of organizations' data and widens customer satisfaction by simplifying routine processes. **Table 4** illustrates the differences between OLTP and OLAP.

4.2. Data Warehouse: Terminologies

The famous approaches used for data modeling are: 1) The Dimensional Model or Star Schema is created using two types of tables namely fact and dimension. 2) The Normalized Model is designed similarly to the way OLTP is designed. Moreover, a star schema is easier and faster in terms of executing queries, while the normalized model is easier when the process of updating information is done [1] [4]. **Table 5** presents a short summary between the dimensional model and the normalized approach in Data Warehouse.

Table 4. The comparison between OLTP and OLAP [20] [21] [22].

	OLTP	OLAP
Organization	It uses workflow per application	It uses dimension and business subject
Data Retention	Short term (2 - 6 months)	Long term (2 - 5 years)
Database Design and Storage	- The design in OLTP is highly normalized. - Gigabytes	- The design is typically de-normalized and contains fewer tables - Terabytes
Data Integration	Minimal or none	High, as part of ETL process
Application	- Real time (short and fast inserts and updates) - write & update - Transactional data - controlling and running fundamental business tasks	- Batch load (includes periodic long-running batch jobs that refresh the data) - Reporting, read-only - Spiked usage - Planning, problem solving and decision support
Adavnatges	- Involves standardized and simple queries that return few records hence, it is faster - A large number of short on-line transaction	- Involves complex queries along with aggregations that return a huge amount of data. - OLAP applications are widely used by Data Mining techniques.

Table 5. The Dimensional model vs the normalized approach in data warehouse.

	The Dimensional Model (star schema or snowflake)	The normalized approach (3NF model)
Data warehouse	States that the data warehouse should be modeled using a Dimensional Model (star schema or snowflake)	States that the data warehouse should be modeled using an E-R model/normalized model
Data	The data is partitioned into either: facts: which are generally numeric transaction data; dimensions: which are the reference information that gives context to the facts.	The data in the data warehouse are stored following database normalization rules. Tables are grouped together by subject areas that reflect general data categories (e.g., data on customers, products, finance, etc.). The normalized structure divides data into entities, which creates several tables in a relational database. Each of the created entities is converted into separate physical tables when the database is implemented.
Advantages	A key advantage of a dimensional approach is that the data warehouse is easier for the user to understand and to use. The retrieval of data from the data warehouse tends to operate very quickly.	The main advantage of this approach is that it is straightforward to add information into the database.
Disadvantages	The main disadvantage of the dimensional approach is that In order to maintain the integrity of facts and dimensions, loading the data warehouse with data from different operational systems is complicated.	A disadvantage of this approach is that, because of the number of tables involved, it can be difficult for users both to join data from different sources into meaningful information and then access the information without a precise understanding of the sources of data and of the data structure of the data warehouse.

4.3. Fundamentals of Data Warehouse

To design the data warehouse model, there are a set of basic fundamentals such as grain, additivity, facts, dimension, and calendar tables. The description of these fundamentals are presented as follow:

- **Grain**: It presents to the scale or level of granularity of fact table such that all facts should have the same grain.
- **Additivity**: This is the property of numeric facts that can be used in competi-

tions such as averaging, min, max, while the other types of facts are also taken into counted.

- **Facts tables**: These tables are connected with dimension tables using one or more foreign keys.
- **Dimension tables**: these types of tables are used to execute the required query and generates the reports.
- **Calendar dimension**: This table is the basic table to make simple dates and is linked with both fact and dimension tables.

The quality of the data warehouse has a significant role in the accuracy of data analysis [23]. The main criteria to measure the quality of a data warehouse is described as follows: 1) Access to information should be easy. 2) Recorded data should be consistent and integrating correctly. 3) Data warehouses should be adapted to any change. Finally, 4) data warehouse should get acceptance by end-users [17].

5. Big Data

In recent years, big data indicates the size, velocity, variety, and diversity of the data sets that are alarmingly growing to create a challenge in storing and analyzing using the conventional database systems. [23]. Recently, many popular data technologies trying to address the challenges of the new Big Data and Internet of Things (IoT) applications that generate data of various sizes are currently ranging from terabytes to petabytes and also considered as a synthetic data generator for structured, semi-structured, and unstructured data [4]. Thus, there are different types of data sources in many domains which create a huge volume of data (Big Data), for instance, video archives, sensor data, Internet text and documents, social networks, tweets, blogs, log files, biochemical, medical records, and transactional records [3]. Recently, the datasets in big data are characterized by n Vs whereby the n refers to the 9 characteristics of the datasets such as Veracity, Variety, Velocity, Volume, Validity, Variability, Volatility, Visualization and Value [24].

5.1. Hadoop

Apache Hadoop is an open-source distributed framework that is widely used for parallel storage, and efficacy in processing the big data on the cluster of machines using high-level programming languages. Hadoop modules have many features to help developers and researchers such as graphical user interfaces, simple administration tools and provide high-level languages such as Java and Python. Moreover, the advantages of this framework are high availability, fault tolerance and scalability to process petabytes of data by the Hadoop cluster. The Hadoop cluster is a group of thousands of computers connected with each other to store Big Data and to run the MapReduce programs in parallel. Generally, users can execute remotely jobs using the Hadoop cluster [25] [26].

Figure 3 shows the principal components of the Hadoop framework in a

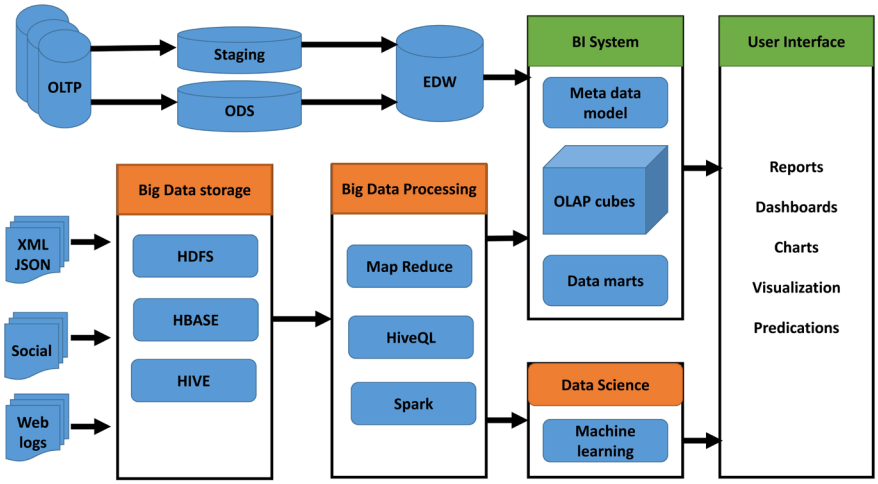

Figure 3. Overview of architecture of data warehouse Hadoop ecosystem components.

master/slave architecture that includes MapReduce, Hadoop Distributed File System (HDFS), YARN, and some other additional modules [27] [28]. Basically, HDFS is considered the primary storage of large datasets in Hadoop and distributed file system management. While, MapReduce is a general programming framework designed specifically for parallel processing the big size of unstructured data as shown in Figure 4.

5.2. HDFS Architecture

The general structure of the HDFS system is simply based on the main communication master/slave framework. The HDFS cluster has two essential elements namely NameNode, and DataNode [29]. Moreover, the cluster has specific NameNode which is responsible for storing metadata, control access to files stored in DataNode and manages the file system. The cluster can have a limited number of DataNodes which are used to store the data as shown in Figure 5. Generally, the big files in HDFS are split into small blocks and stored in a specific number of Datanodes on the cluster. The default size of blocks is a little bit large (64 MB or 128 MB), to decrease the running time. In general, the replication factor determines how many copies of one block are saved in different Datanodes. If the replication factor is 2 the block should be on two Datanodes. Figure 5 demonstrates a simple example of large files that are stored in the Namenode and four DataNodes. The main advantages of HDFS are 1) it is developed to be more scalable and has the capability of fault tolerance, 2) it saves big files for future use, these files split into blocks which replicated on two or more DataNodes, 3) the performance of HDFS is scaled by increasing the number of Data Nodes [30].

5.3. MapReduce

Recently, Dean and Ghemawat developed the most important programming model MapReduce which is widely used for the Cloud computing framework

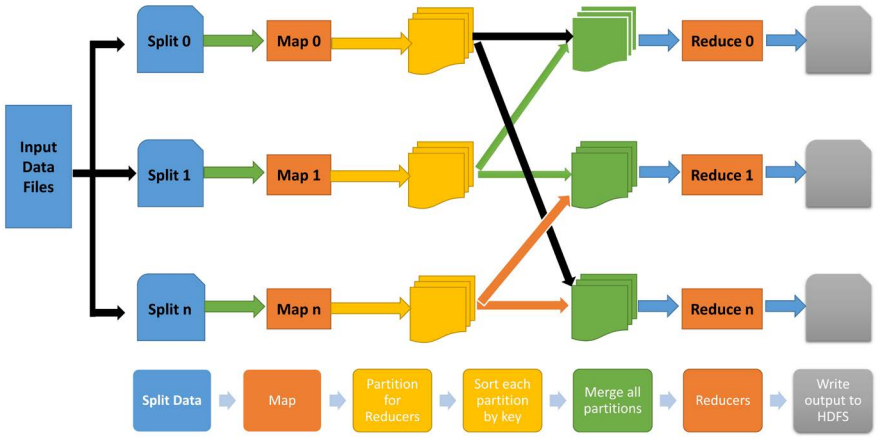

Figure 4. Process flow in Hadoop [5].

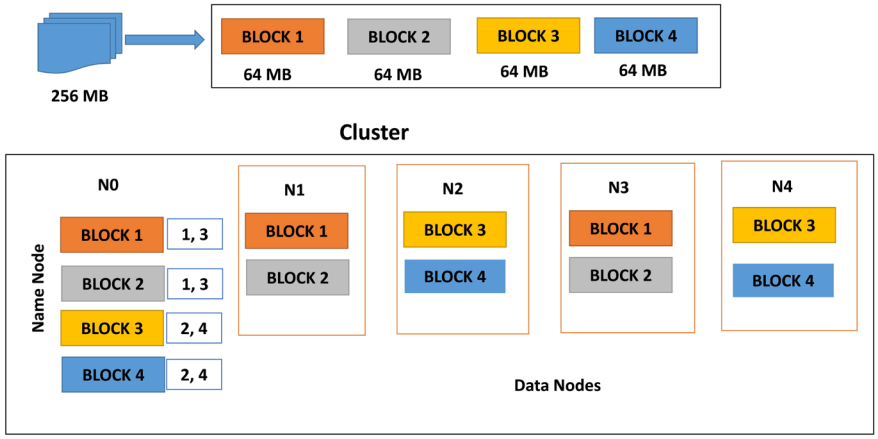

Figure 5. HDFS cluster in Hadoop [5].

and is especially supported by Google under the Apache Hadoop project [31]. In general, the developers address big issues by writing the MapReduce programs where the input files split into small chunks to adapt to the HDFS system and the parallel computations [4].

The "Map" and "Reduce" classes are the fundamental components of MapReduce. **Figure 6** shows the main steps involved in the process flow of MapReduce. In the Map step, the input data divides into a number of small chunks such that all sub-files are allocated parallelly to different mappers. Moreover, the basic idea of the mapper method is to read the corresponding chunk as a bench of keys and their value pairs and similarly, the results of this method are a set of (key, value) pairs. The shuffles and sorts phase comes immediately after the map phase and the input for this stage is the result of the mapper functions, and it produces the (key, value) pairs that assigned to the reducers. In the final process, the output (key, value) pairs of previous tasks are grouped based on the key and assigned to the reducers to produce the final results which are stored in HDFS [32]. The Hadoop framework has a JobTracker service that assigns the tasks of MapReduce to clearly defined nodes within the Hadoop cluster, specifically the nodes that have the data. On the other hand, the TaskTracker is a node that accepts

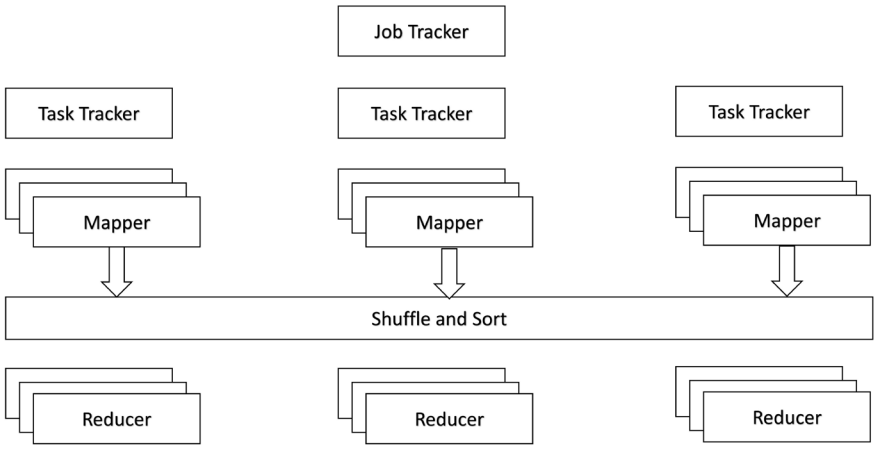

Figure 6. JobTracker and TaskTracker in Hadoop [5].

operations from a JobTracker service such as Map, Reduce and Shuffle. During the execution of MapReduce job, JobTracker and TaskTracker tools are applicable to schedule, monitor, and restart processes in case of failing as shown in Figure 6.

5.4. Hadoop Ecosystem

This section describes the Hadoop Ecosystem that has many modules, MapReduce and HDFS that are used in the ETL operations [33]. Moreover, Hadoop has different software for particular tasks such as Apache Spark, Apache Hive are used for data processing and Apache Oozie, Apache Drill for jobs orchestration, while Apache Spark and Apache Mahout are applied to build high-performance machine learning models [34].

1) Apache Sqoop

It is a command-line-based tool and an open-source under the Apache software, that allows for efficient importing of records from relational database tables to HDFS directories as database tables in the Hadoop framework. Therefore, the essential tasks of Sqoop are transferring stored data from Oracle or MySQL database into the HDFS database, then the MapReduce program executed on the imported data, and finally exports data in database tables or Data warehouse [3] [35] [36]. There are many versions of Hadoop connectors provided by famous EDW vendors such as IBM Db2 [37], HP Vertica [38] and Oracle [39] bulk-loads data from Hadoop to Oracle Database.

2) Apache Hive

This is an open-source project and an efficient query language that simplifies the development of applications using the MapReduce framework. Apache Hive has two main components namely Hive Query Language (HiveQL) and Hive metastore [3] [40] [41]. HiveQL is a query language for Hive software and used to facilitate writing queries data stored in Apache HBase and HDFS. While, the Hive metastore is the master repository of Hive metadata which is used to store metadata about data files, blocks in the HDFS NameNode [42]. Moreover, Hive

is used in data warehousing implementation such that it facilitates the operations such as reading, writing, and managing large files stored in HDFS [43].

3) ODBC/JDBC Connectors

Basically, any version of Apache Hadoop software provides a JDBC/ODBC connectors for HBase and Hive and interface to make a connection with Business Intelligence and different visualization tools. Apache Hive ODBC and Hive JDBC drivers are software components used for translating from traditional SQL queries into HiveQL commands so that it can execute upon the data in the Apache Hadoop/Hive distributions [3] [44].

6. Implementation

The commonly used methods for building a meteorological data warehouse are the classical DW, big data tools such as Hadoop, Sqoop, Hive, Pig, and Spark. Finally, the hybrid DW model merges traditional DW with big data. In this work, the proposed data warehouse model constructed based on the third approach such that the following big data tools are applied to implement the model as shown in Figure 7. The list of tools of big data are described as follows:

- **RDBMS**: It is used to store the records of the collected data.
- **SQOOP**: It is used to import records from RDBMS into HDFS, and to transfer the final results of aggregation operations to the data warehouse.
- **HDFS**: It is used to save big files.
- **Hive**: It is used to create tables and databases on HDFS. Moreover, it is has the ability to perform join, partition, merge, or aggregation operations.

6.1. High Dimensional Schema for NCDC Dataset

The data-warehouse schema is a logical description of the whole database [8]. The main components of data-warehouse include fact tables, dimension tables, and their dependencies. A dimension is simply a row and column of the high-dimensional table that contains the number of samples and their corresponding

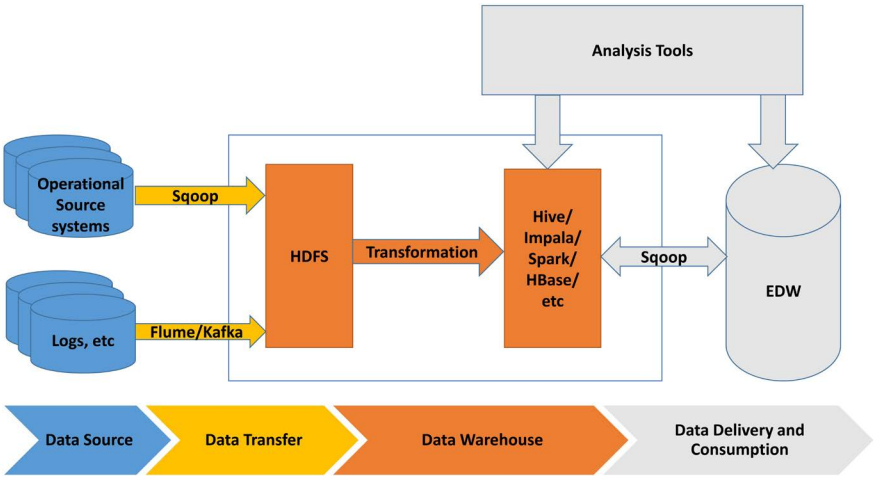

Figure 7. High level of Hadoop data warehouse.

attributes. There are some main operations that can be applied to the high-dimensional table such as grouping, filtering and labeling. Moreover, the samples in dimension tables are identified by a unique Key and each column represents a range of values that are found by measuring using given units.

The available data set is collected for each year as compressed files. Figure 8 shows the design of the relational schema for the collected data. For instance, the weather_station table contains information about each station that has columns namely station_STN, latitude, longitude, ..., etc. Similarly, the countries table has information about each country and it is linked to the states table. Finally, each parameter key in the Parameter table is used to connect with the corresponding table.

Figure 9 demonstrates the proposed star schema on NCDC weather datasets that contains two types of tables and it presents one fact table and 7-dimensional tables. The primary task of OLTP schema is used for performing the preprocessing stage such as reduce redundancy, normalize the data and check for its integrity. In addition, it also has another advantage such as it is possible to create, update or delete any column in a particular table. The weather fact table has seven dimension tables, namely STATION, TIME, TEMPERATURE, PARAMETER, SLP, DEWP, and PRCP. In general, the fact table consists of a number of primary keys and many keys that refer to their corresponding dimension table. On the other hand, the primary key in the dimension table has the same name as the table that contains observations collected from a particular station. Table 6 describes the dimension tables including all the relevant attributes.

Table 6. The dimension tables and attributes in OLTP Schema.

TABLE	Attributes
Countries	Country_key, Country_Name, Short_Name
States	State_key, Long_Name, Short_Name, Country_key
Cities	City_key, City_Name, Short_Name, State_key
Stations	Station_STN, Station_wban, Station_name, Country, call_st, latitude, longitude, Elevation, BEGIN_date, END_date
Parameters	Parameter_key, Station_STN, Station_wban, Yearmody, Temperature, DEWP, SLP, STP, VISIB, WDSP, MXSPD, GUST, MAX, MIN, PRCP, SNDP, I_FOG, I_RAIN_DZL, I_SNW_ICE, I_HAIL, I_THUNDER, I_TDO_FNL
Daily_Measures_Temperature	Temperature_key, Temperature_CNT, Month, Year, Day1, Day2, Day3, ..., Day31
Daily_Measures_DEWP	DEWP_key, DEWP_CNT, Month, Year, Day1, Day2, Day3, ..., Day31
Daily_Measures_SLP	SLP_key, SLP_CNT, Month, Year, Day1, Day2, Day3, ..., Day31
Daily_Measures_STP	STP_key, STP_CNT, Month, Year, Day1, Day2, Day3, ..., Day31
Daily_Measures_VISIB	VISIB_key, VISIB_CNT, Month, Year, Day1, Day2, Day3, ..., Day31
Daily_Measures_WDSP	WDSP_key, WDSP_CNT, Month, Year, Day1, Day2, Day3, ..., Day31
Daily_Measures_MXSPD	MXSPD_key, Month, Year, Day1, Day2, Day3, ..., Day31
Daily_Measures_GUST	GUST_key, Month, Year, Day1, Day2, Day3, ..., Day31
Daily_Measures_PRCP	PRCP_key, PRCP_FLAG, Month, Year, Day1, Day2, Day3, ..., Day31

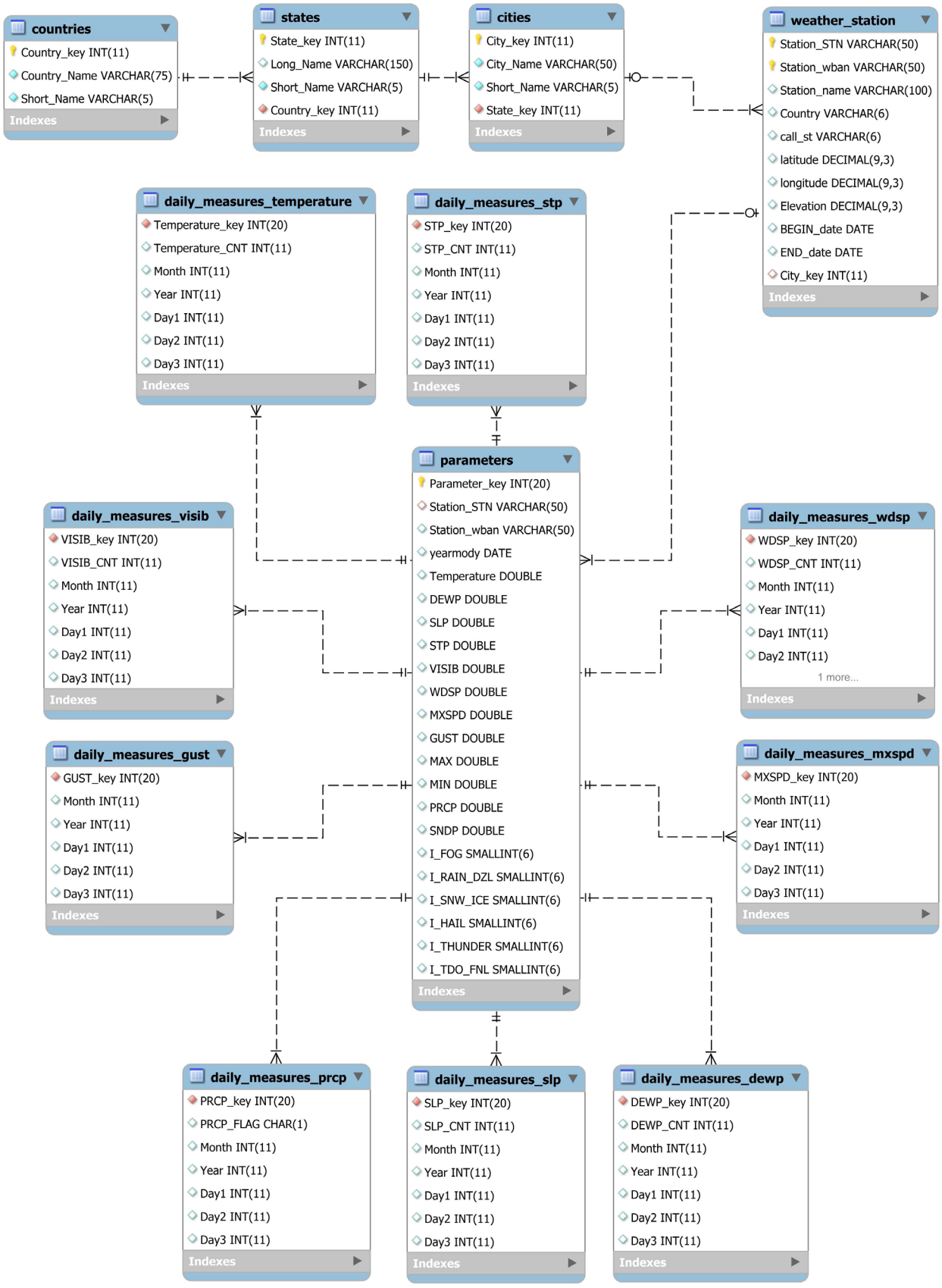

Figure 8. Relational schema for NCDC data set.

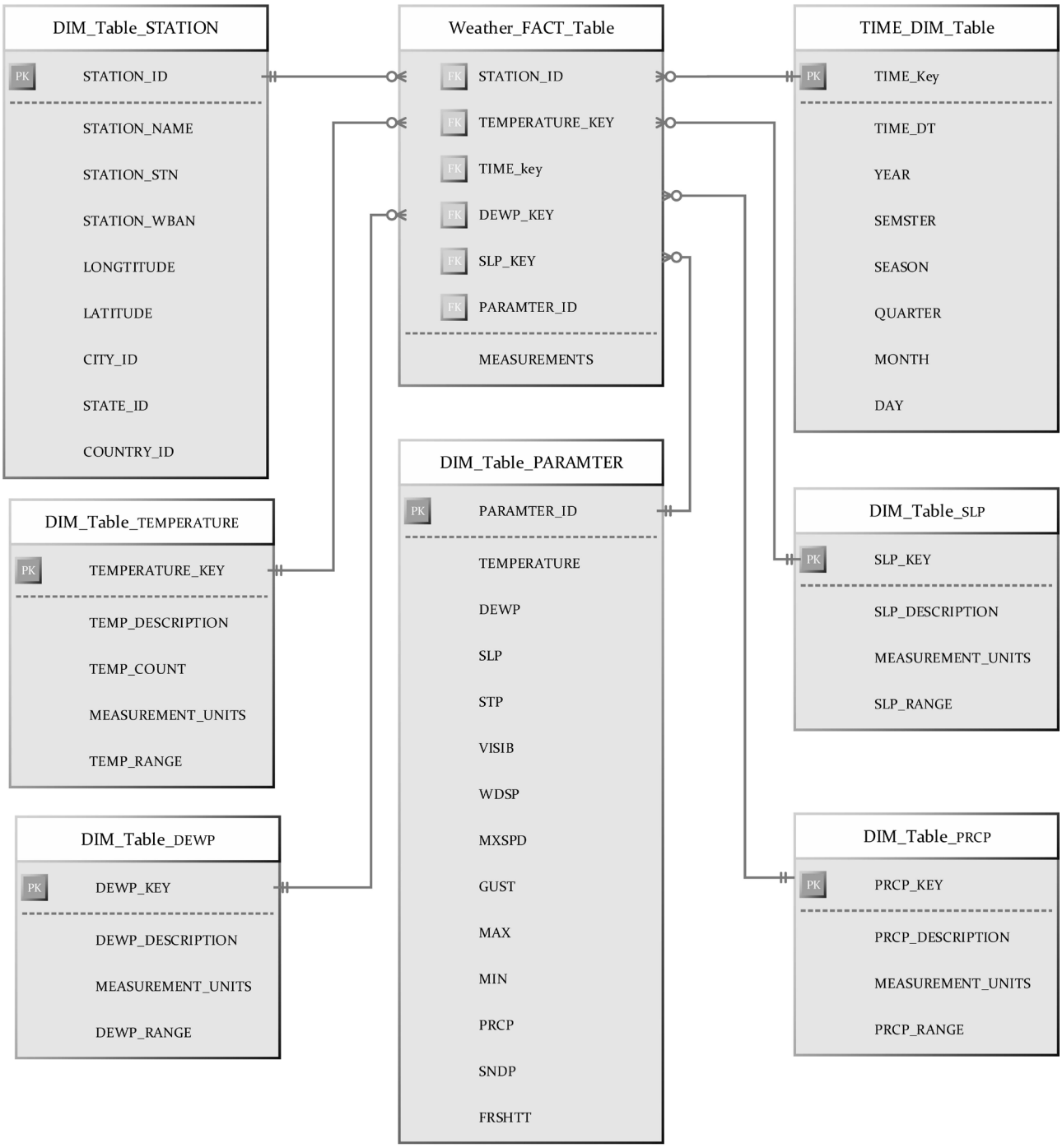

Figure 9. A part of data warehouse for weather data set.

6.2. The Proposed Hadoop Data Warehouse Model

Figure 10 presents the high level conceptual model for Hadoop data warehouse. The first step in the building of such a model is to import data from OLTP tables to HDFS. For instance, the weather_fact, parameter_fact and the rest of the tables are imported using the Sqoop tool. The Sqoop job for any table from OLTP imports all data and overwrite the existing old table. To solve this problem,

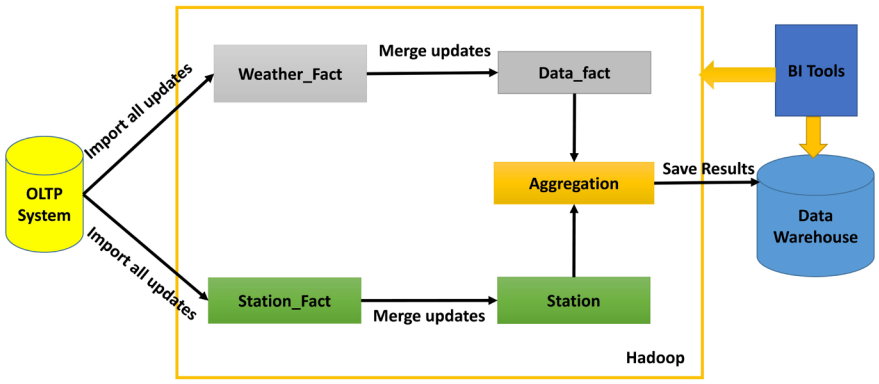

Figure 10. High level of the proposed Hadoop data warehouse model.

only the import jobs for each table transfer new records and consequently updates only the current records. Finally, the aggregation operations such as max, min, average, and count are done for the new version of the updated tables.

Figure 11 illustrates the simplified design of the proposed Hadoop data warehouse model. This design composes of three main tables namely Countries, Parameters, and Weather_Stations. However, the other tables from OLTP schema have been denormalized into one of these three main tables, for instance, the daily_measure_VISIB, daily_measure_PRCP, daily_measure_DEWP, and daily_measure_SLP tables have been denormalized into Parameters.

There are many reasons to use the Hadoop framework in the data warehouse model, it has the capability to store the same records such as OLTP, save large files that can be distributed over HDFS and both DW and Hadoop are allowed the data to be moved between them efficiently. However, the way to store data in Hadoop is totally different from OLTP. In OLTP, it is allowed to update data by one record at a time and only one operation is executed such as insert/update/delete. So, the OLTP schema allows the following operations for the values of the records: updates/deletes/inserts to change these values. On the other hand, for the HDFS system in Hadoop, it is not allowed to update the values of records, it allows only deleting old data and replaces it by the new ones. To solve this problem in Hadoop, it is essential to create two versions of tables such that the second table is used to append only new records. For instance, the weather_stations table has another version Station_history which is an append-only table and is used to save the history of all stations. Sequentially, once the import job executed, the new data imported from OLTP appended to the end of the Station_history table and the final data are stored in weather_stations.

To make this clear, the following example illustrates the concept of insert/update/delete in Hadoop, Table 7 presents a sample of Stations table in the OLTP.

The same data of Stations table are stored in Hadoop as shown in Table 8.

In case of any update operation, it is done in the Weather_stations table such as an add a new station, so the data of this table needs to be updated in Hadoop and have only full records. Thus, the HDFS system has two versions of the same

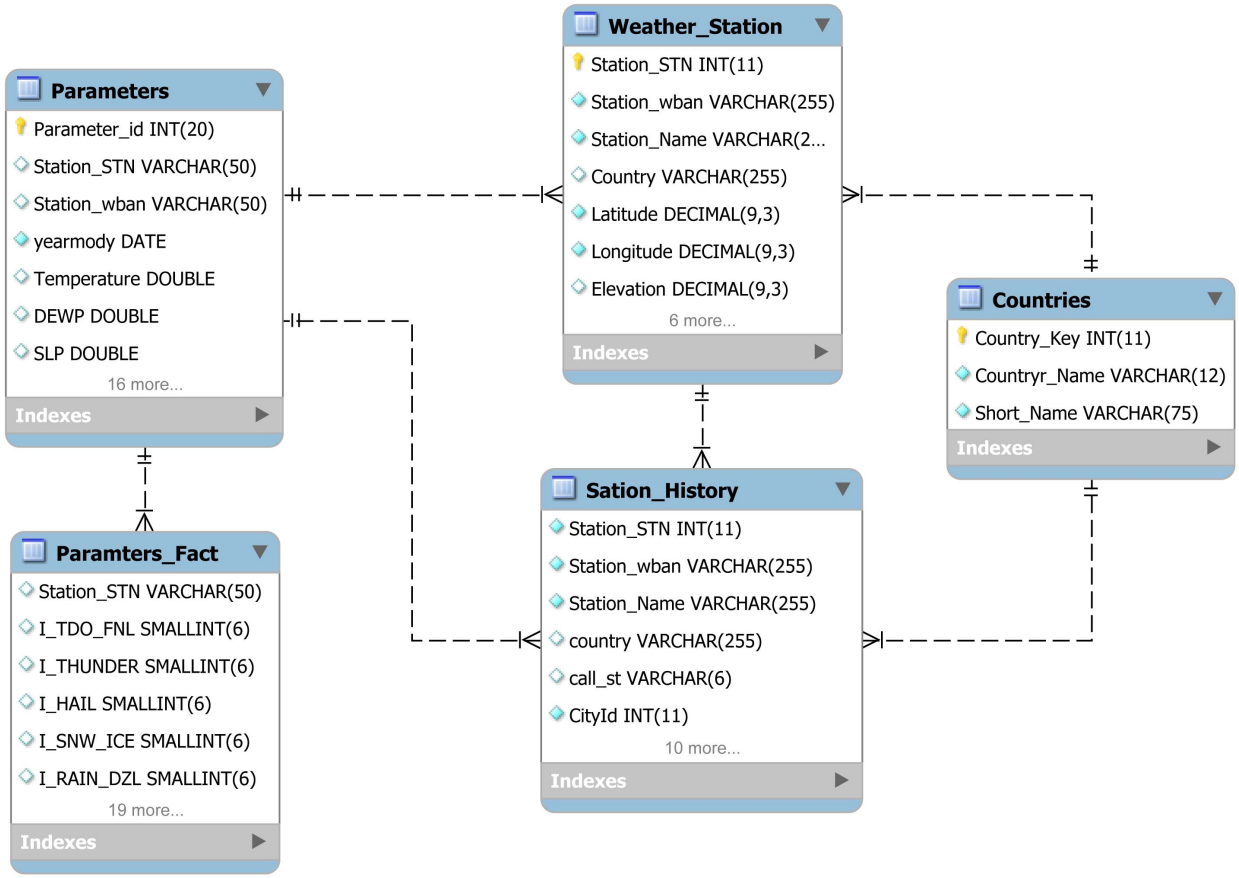

Figure 11. Hadoop data warehouse model for weather data set.

Table 7. A sample of rows of station table in OLTP database.

USAF	WBAN	STATION NAME	CTRY	LAT	LON	ELEV(M)	BEGIN	END
410090	99999						19851001	20110614
410060	99999	MUWAIH	SA	+22.433	+041.750	+0971.0	19851001	20110614
410080	99999	ZULM	SA	+22.717	+042.167	+0870.0	20041015	20041015
410100	99999	LAYLA	SA	+22.333	+046.733	+0543.0	19910408	20030329
410140	99999	OBAYLAH (AUT)	SA	+22.217	+050.883	+0588.0	19830201	20030315
410160	99999	SHAWALAH	SA	+22.517	+054.050	+0468.0	19830402	20050526

Table 8. Stations table in Hadoop before update.

USAF	WBAN	STATION NAME	CTRY	LAT	LON	ELEV(M)	BEGIN	END
410090	99999						19851001	20110614
410060	99999	MUWAIH	SA	+22.433	+041.750	+0971.0	19851001	20110614
410080	99999	ZULM	SA	+22.717	+042.167	+0870.0	20041015	20041015
410100	99999	LAYLA	SA	+22.333	+046.733	+0543.0	19910408	20030329
410140	99999	OBAYLAH (AUT)	SA	+22.217	+050.883	+0588.0	19830201	20030315
410160	99999	SHAWALAH	SA	+22.517	+054.050	+0468.0	19830402	20050526

table in order to address the issue of update data in HDFS. On the other hand, Station_history includes all the imported records from the same version of this table in the OLTP schema as shown in Table 8. Similarly, Table 9 demonstrates the final version of Weather_stations table that has only the complete rows.

6.3. The Aggregates Design

The primary task of a data warehouse is to make great flexibility and efficiency for the query process. So, aggregations are used to decrease the query time using pre-computed summary data. In this section, the following tasks namely ingestion, aggregation, and data export are briefed.

1) Ingestion

The first stage of the conventional ETL approach is to transfer the stored data from the OLTP schema to a data warehouse. Generally, the Sqoop tool is used for ingesting data from OLTP into HDFS in Hadoop as shown in Figure 12. Moreover, to ingest a small data from OLTP into Hadoop is require one task using Sqoop. On the other hand, if the data size is large, then it will need many tasks or repeat the Sqoop job many times as shown in Figure 13. Moreover, the list of files transferred after Sqoop task done is shown in Figure 14.

2) Aggregation

The aggregation operation is one of the most time-consuming operations to perform in the classical database. Thus, to reduce the running time of ETL and aggregation operations are implemented in Hadoop. As the Hadoop framework

Table 9. Stations table in Hadoop after update.

USAF	WBAN	STATION NAME	CTRY	LAT	LON	ELEV(M)	BEGIN	END
410060	99999	MUWAIH	SA	+22.433	+041.750	+0971.0	19851001	20110614
410080	99999	ZULM	SA	+22.717	+042.167	+0870.0	20041015	20041015
410100	99999	LAYLA	SA	+22.333	+046.733	+0543.0	19910408	20030329
410140	99999	OBAYLAH (AUT)	SA	+22.217	+050.883	+0588.0	19830201	20030315
410160	99999	SHAWALAH	SA	+22.517	+054.050	+0468.0	19830402	20050526

```
User@Root        ~$ sqoop import-all-tables \
>   -m 12 \
>   --connect "jdbc:mysql://localhost:3306/weather" \
>   --username=hive_user \
>   --password=hive_password \
>   --as-textfile \
>   --warehouse-dir=/user/hive/warehouse/data
Warning: /usr/local/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/local/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
20/06/17 02:51:25 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
20/06/17 02:51:25 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
20/06/17 02:51:25 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
20/06/17 02:51:26 INFO tool.CodeGenTool: Beginning code generation
20/06/17 02:51:26 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `Cities` AS t LIMIT 1
20/06/17 02:51:26 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `Cities` AS t LIMIT 1
```

Figure 12. Sqoop job for transfer data from Mysql into HDFS in Hadoop.

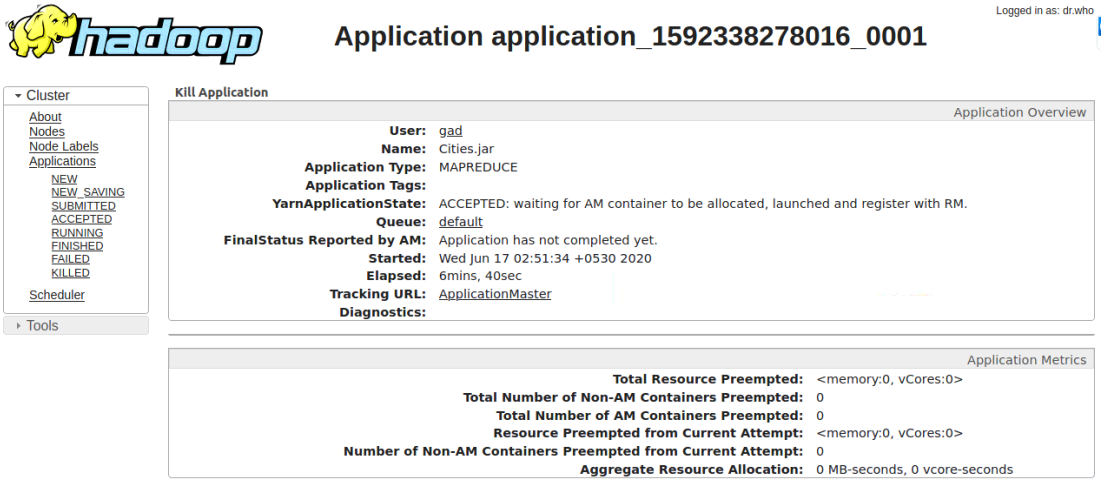

Figure 13. The description of Sqoop job that used for transfer data into Hadoop.

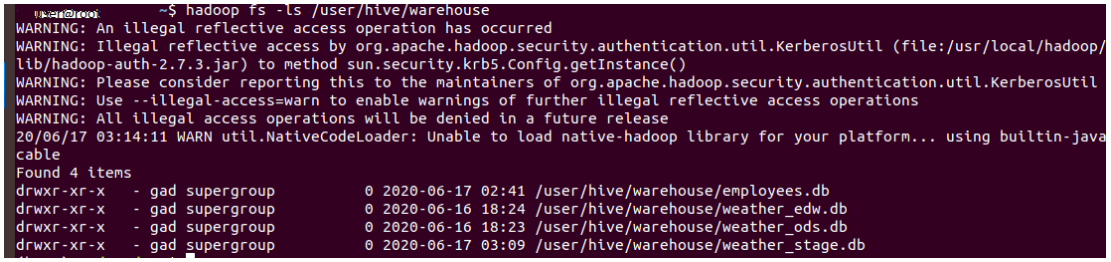

Figure 14. The list of files transferred after Sqoop task done.

is distributed and can be run tasks in parallel, the aggregation tasks are executed faster than the OLTP database. Hive and Impala are the most popular tools known for aggregation over Hadoop. Finally, aggregation operations such as average, max, count, and summation are implemented cheaper and scalable using Hadoop that store a huge number of records. For example, the following Hive code calculates the records count and the average temperature for each station, respectively.

```
CREATE TABLE station_record_count AS SELECT station_STN, station_wban, COUNT(*) AS count FROM Parameters_fact GROUP BY station_STN, station_wban
```

```
SELECT station_STN, yearmody, ROUND(AVG(temperature), 1) AS temperature FROM Parameters WHERE yearmody >= date_sub('2019-01-03', 7) AND yearmody <= date_sub('2019-01-12', 1) GROUP BY station_STN, yearmody limit 5;
```

3) Data Export

This is the next step after ingestion and aggregation to transfer the cleaned data from HDFS to the real data warehouse. In this stage, the preferred tool is Sqoop that can be applied to shift the data from the Hadoop system to the traditional data warehouse with insert and update operations. For instance, the following Sqoop code exports data from avg_temp table in weather_dwh database stored in Hadoop to a database table.

```
sqoop export --connect \
jdbc::mysql://localhost:3306/weather_dwh \
--username hive_user \
--table avg_temp
--export-dir /user/hive/warehouse/avg_temp \
-m 16 --update-key station_STN \
--input-fields-terminated-by '\001' \
--lines-terminated-by '\n
```

7. Conclusion

In this work, we explored the different notions regarding big data, data warehouse, and presented in detail the proposed data warehouse for weather data that typically constructed on top of the strong Hadoop system. Moreover, the flexible meteorological data warehouse successfully produced using the suggested star schema model and different Big Data software. The presented schema includes all necessary fact and dimension tables in order to deal with scale and efficient analytical models. Furthermore, the suggested data warehouse model optimized for NCDC that was available. The advantages of this model are the possibility to add new variables; different queries are easily done in a flexible way, and similarly, it easy to extract data. Finally, the proposed model is flexible, adaptable, and quite qualified for the rapid increase of data from different weather variables without any major change.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Doreswamy, I. and Manjunatha, B.R. (2017) Hybrid Data Warehouse Model for Climate Big Data Analysis. 2017 *International Conference on Circuit, Power and Computing Technologies* (*ICCPCT*) *IEEE*, Kollam, 20-21 April 2017. https://doi.org/10.1109/ICCPCT.2017.8074229

[2] N. Climatic Data Center (NCDC) (2016) Noaa's National Centers for Environmental Information (NCEI). https://www.ncdc.noaa.gov/data-access/land-based-station-data/land-based-datasets

[3] I. Corporation (2013) Extract, Transform, and Load Big Data with Apache Hadoop. *Intel. Corporation, Vol. White Paper Big Data Analytics*, 1-5. https://software.intel.com/content/www/us/en/develop/download/white-paper-extract-transform-and-load-big-data-with-apache-hadoop.html

[4] Amr Awadallah, D.G. (2013) Hadoop and the Data Warehouse: When to Use Which. Cloudera Corporation and Teradata Corporation, Vol. White Paper, 1-19. https://kannandreams.files.wordpress.com/2013/10/hadoop-use-case-1.pdf

[5] Alexander, I., Rassetiadi, R., Garcia, S., Girsang, A.S. and Isa, S.M. (2018) Business Solution for Choosing Products Using Data Warehouse in Payment Solution. 2018 *Indonesian Association for Pattern Recognition International Conference*, Jakarta, 7-8 September 2018. https://doi.org/10.1109/INAPR.2018.8627028

[6] Chen, D.-Q., Wang, W.-Y. and Yang, H.-K. (2010) Application Research on Data

Warehouse of Hydrological Data Comprehensive Analysis. 2010 3*rd International Conference on Computer Science and Information Technology*, Vol. 9, 140-143. https://doi.org/10.1109/ICCSIT.2010.5565123

[7] Kalra, G. and Steiner, D. (2005) Weather Data Warehouse: An Agent-Based Data Warehousing System. *Proceedings of the* 38*th Annual Hawaii International Conference on System Sciences*, Big Island, 6 January 2005.

[8] Ma, N., Yuan, M., Bao, Y., Jin, Z. and Zhou, H. (2010) The Design of Meteorological Data Warehouse and Multidimensional Data Report. 2010 *Second International Conference on Information Technology and Computer Science*, Kiev, 24-25 July 2010, 280-283. https://doi.org/10.1109/ITCS.2010.75

[9] Tian, Y. (2019) Hybrid Data Warehouse. In: *Encyclopedia of Big Data Technologies*, Springer International Publishing, Berlin, 979. https://doi.org/10.1007/978-3-319-77525-8_100167

[10] Zhang, Z. and Li, J. (2020) Big Climate Data. In: *Big Data Mining for Climate Change*, Elsevier, Amsterdam, 1-18. https://doi.org/10.1016/B978-0-12-818703-6.00006-4

[11] Doreswamy and Harishkumar, K. (2018) Multidimensional Data Model for Air Pollution Data Analysis. 2018 *International Conference on Advances in Computing, Communications and Informatics* (*ICACCI*) *IEEE*, Bangalore, 19-22 September 2018. https://doi.org/10.1109/ICACCI.2018.8554621

[12] Chen, S. (2010) Cheetah: A High Performance, Custom Data Warehouse on Top of Mapreduce. *Proceedings of the VLDB Endowment*, **3**, 1459-1468. https://doi.org/10.14778/1920841.1921020

[13] Dimri, P. and Gunwant, H. (2012) Conceptual Model for Developing Meteorological Data Warehouse in Uttarakhand—A Review. *Journal of Information and Operations Management*, **3**, 107-110.

[14] José Torres-Jiménez, J.F.G. (2004) A Data Warehouse for Weather Information: A Pattern Recognition Solution for Climatic Conditions in México. *Proceedings of the Sixth International Conference on Enterprise Information Systems*, **1**, 562-565.

[15] Wadkar, S. and Siddalingaiah, M. (2014) Data Warehousing Using Hadoop. In: *Pro Apache Hadoop*, Apress, New York, 217-239. https://doi.org/10.1007/978-1-4302-4864-4_10

[16] Duque-Méndez, N.D., Orozco-Alzate, M. and Vélez, J.J. (2014) Hydro-Meteorological Data Analysis Using OLAP Techniques. *DYNA*, **81**, 160-167. https://doi.org/10.15446/dyna.v81n185.37700

[17] Vuong, N.-A.L.-K., Ngo, M. and Kechadi, M.-T. (2018) An Efficient Data Warehouse for Crop Yield Prediction. *Proceedings of the* 14*th International Conference on Precision Agriculture*, Montreal, 24-27 June 2018. https://researchrepository.ucd.ie/bitstream/10197/10118/2/1807.00035v1.pdf

[18] Narayan, R. and Mehta, G. (2020) Design of Customer Information Management System. In: *Data Communication and Networks*, Springer, Berlin, 195-216. https://doi.org/10.1007/978-981-15-0132-6_13

[19] Jin, Z.-H., Shi, H., Hu, Y.-X., Zha, L. and Lu, X. (2020) Cirrodata: Yet Another Sql-on-Hadoop Data Analytics Engine with High Performance. *Journal of Computer Science and Technology*, **35**, 194-208. https://doi.org/10.1007/s11390-020-9536-z

[20] Conn, S. (2005) OLTP and OLAP Data Integration: A Review of Feasible Implementation Methods and Architectures for Real Time Data Analysis. *Proceedings. IEEE Southeast Con*, Ft. Lauderdale, 8-10 April 2005, 515-520.

[21] Chohan, M.A. and Javed, M.Y. (2010) OLAP and OLTP Data Integration for Oper-

ational Level Decision Making. 2010 *International Conference on Networking and Information Technology*, Manila, 11-12 June 2010, 493-496. https://doi.org/10.1055/s-0030-1258076

[22] Giceva, J. and Sadoghi, M. (2019) Hybrid OLTP and OLAP. In: *Encyclopedia of Big Data Technologies*, Springer International Publishing, Berlin, 979-986. https://doi.org/10.1007/978-3-319-77525-8_179

[23] Aftab, U. and Siddiqui, G.F. (2018) Big Data Augmentation with Data Warehouse: A Survey. 2018 *IEEE International Conference on Big Data* (*Big Data*), Seattle, 10-13 December 2018, 2785-2794. https://doi.org/10.1109/BigData.2018.8622206

[24] Owais, S.S. and Hussein, N.S. (2016) Extract Five Categories CPIVW from the 9v's Characteristics of the Big Data. *International Journal of Advanced Computer Science & Applications*, **1**, 254-258. https://doi.org/10.14569/IJACSA.2016.070337

[25] Dean, J. and Ghemawat, S. (2008) Mapreduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, **51**, 107-113. https://doi.org/10.1145/1327452.1327492

[26] Nazari, E., Shahriari, M.H. and Tabesh, H. (2019) Big Data Analysis in Healthcare: Apache Hadoop, Apache Spark and Apache Flink. *Frontiers in Health Informatics*, **8**, 14. https://doi.org/10.30699/fhi.v8i1.180

[27] Anthony, B., *et al.* (2016) Ecosystem at Large: Hadoop with Apache Bigtop. In: *Professional Hadoop*, John Wiley & Sons Inc., Hoboken, 141-160. https://doi.org/10.1002/9781119281320.ch7

[28] Vavilapalli, V.K., Seth, S., Saha, B., Curino, C., O'Malley, O., Radia, S., Reed, B., Baldeschwieler, E., Murthy, A.C., Douglas, C., Agarwal, S., Konar, M., Evans, R., Graves, T., Lowe, J. and Shah, H. (2013) Apache Hadoop YARN. In: *Proceedings of the* 4*th annual Symposium on Cloud Computing*, ACM Press, New York. https://doi.org/10.1145/2523616.2523633

[29] Tian, Y., Özcan, F., Zou, T., Goncalves, R. and Pirahesh, H. (2016) Building a Hybrid Warehouse: Efficient Joins between Data Stored in HDFS and Enterprise Warehouse. *ACM Transactions on Database Systems*, **41**, 1-38. https://doi.org/10.1145/2972950

[30] Pang, G. and Li, H. (2018) Caching for SQL-on-Hadoop. In: *Encyclopedia of Big Data Technologies*, Springer International Publishing, Berlin, 1-5. https://doi.org/10.1007/978-3-319-63962-8_249-1

[31] Dean, J. and Ghemawat, S. (2008) MapReduce. *Communications of the ACM*, **51**, 107. https://doi.org/10.1145/1327452.1327492

[32] Sakr, S. and Zomaya, A.Y. (2019) Apache Hadoop. In: *Encyclopedia of Big Data Technologies*, Springer International Publishing, Berlin, 58. https://doi.org/10.1007/978-3-319-77525-8_100009

[33] Koitzsch, K. (2017) Pro Hadoop Data Analytics. Apress, New York. https://doi.org/10.1007/978-1-4842-1910-2

[34] Elahi, I. (2019) Hello Apache Spark. In: *Scala Programming for Big Data Analytics*, Apress, New York, 261-299. https://doi.org/10.1007/978-1-4842-4810-2_14

[35] Ting, K. and Cecho, J.J. (2013) Apache Sqoop Cookbook. O'Reilly Media, Inc., Sebastopol.

[36] Vohra, D. (2016) Apache Sqoop. In: *Practical Hadoop Ecosystem*, Apress, New York, 261-286. https://doi.org/10.1007/978-1-4842-2199-0_5

[37] Özcan, F., Hoa, D., Beyer, K.S., Balmin, A., Liu, C.J. and Li, Y. (2011) Emerging Trends in the Enterprise Data Analytics. In: *Proceedings of the* 2011 *International*

Conference on Management of Data, ACM Press, New York. https://doi.org/10.1145/1989323.1989446

[38] Vertica (2016) Hadoop Integration Guide—Hp Vertica Analytic Database. https://softwaresupport.softwaregrp.com/doc/KM00681126?fileName=hp_man_Vertica_7.0.x_Hadoop_integration_pdf.pdf

[39] Oracle (2012) High Performance Connectors for Load and Access of Data from Hadoop to Oracle Database. https://www.oracle.com/technetwork/bdc/hadoop-loader/connectors-hdfs-wp-1674035.pdf

[40] Huai, Y., Zhang, X., Chauhan, A., Gates, A., Hagleitner, G., Hanson, E.N., Malley, O.O., Pandey, J., Yuan, Y. and Lee, R. (2014) Major Technical Advancements in Apache Hive. In: *Proceedings of the* 2014 *ACM SIGMOD International Conference on Management of Data*, ACM Press, New York. https://doi.org/10.1145/2588555.2595630

[41] Thusoo, A., Sarma, J.S., Jain, N., Shao, Z., Chakka, P., Anthony, S., Liu, H., Wyckoff, P. and Murthy, R. (2009) Hive: A Warehousing Solution over a Map-Reduce Framework. *Proceedings of the VLDB Endowment*, **2**, 1626-1629. https://doi.org/10.14778/1687553.1687609

[42] Vohra, D. (2016) Apache Hive. In: *Practical Hadoop Ecosystem*, Apress, New York, 209-231. https://doi.org/10.1007/978-1-4842-2199-0_3

[43] Oracle (2017) Apache Hive SQL Conformance (2017). https://cwiki.apache.org/confluence/display/Hive/Apache+Hive+SQL+Conformance

[44] Khalifa, S. (2018) Tools and Libraries for Big Data Analysis. In: *Encyclopedia of Big Data Technologies*, Springer International Publishing, Berlin, 1-7. https://doi.org/10.1007/978-3-319-63962-8_282-1