A distributed data processing platform over meteorological big data using MapReduce

Tao Huang and Shengjun Xue*

School of Computer Science and Technology,
Silicon Lake College,
Suzhou, China
Email: nuisthuangtao@163.com
Email: sjxue@163.com
*Corresponding author

Xiang Li

School of Computer and Software,
Nanjing University of Information Science and Technology,
Nanjing, China
Email: nuistlixiang@gmail.com

Feng Luo

Meteorological Station,
Shanghai Jiading Meteorological Service,
Intersection of Shisheng Road and Shengzhu Road,
Jiading 201800, China
Email: lffpo2019@163.com

Abstract: In the era of big data, the data of the meteorological departments grows explosively, which puts higher requirements on the real-time processing of meteorological big data. Besides, the efficient storage for the massive meteorological data has also attracted much attention from the meteorological departments. Therefore, in response to the urgent requirements of meteorological big data in processing and storage, a distributed data processing platform over meteorological big data using MapReduce is designed. Technically, the platform develops corresponding real-time strategies according to various data properties, and obtains meteorological big data in real-time from multiple channels. Based on the MapReduce real-time processing, we realise the distributed storage to store the meteorological big data in the platform. Overall, our platform improves the real-time, reliability, availability and the access efficiency of meteorological big data which is easy to expand and also has a good reference value for big data processing in other similar industries.

Keywords: meteorology; massive; big data; data processing; cloud; platform; distributed; MapReduce; real-time; sharing.

Reference to this paper should be made as follows: Huang, T., Xue, S., Li, X. and Luo, F. (2019) 'A distributed data processing platform over meteorological big data using MapReduce', *Int. J. Intelligent Internet of Things Computing*, Vol. 1, No. 1, pp.74–85.

Biographical notes: Tao Huang received his Bachelors and Masters from the School of Computer and Software, Nanjing University of Information Science and Technology, in 2010 and 2013 respectively. He worked in the Shanghai Jiading District Meteorological Bureau from 2013 to 2018, and started his career as a Lecturer at the School of Computer and Technology at Silicon Lake College in August 2018. His research interests include issues related to artificial intelligence algorithms, big data and cloud computing.

Shengjun Xue graduated from the Zhejiang University in 1983 with a major in Computer Application. He was promoted to Professor and was hired as a Doctoral Supervisor, in 2000 and 2012 respectively. He is currently an academic leader in high performance computing at the School of Computer and Technology at Silicon Lake College. His research interests include issues related to big data and cloud computing.

Xiang Li is currently working towards his Bachelors in Computer Science and Technology at the School of Computer and Software in Nanjing University of Information Science and Technology. His research interests include mobile cloud computing and edge computing.

Feng Luo received his Masters from the School of Atmospheric Physics, Nanjing University of Information Science and Technology in 2015. He has worked in the Shanghai Jiading District Meteorological Bureau since 2015. His research interests include issues related to the atmospheric physics and the assimilation of meteorological data.

1 Introduction

With the development of information technology, there are an increasing amount of professional equipment for observing meteorological elements from space, observation stations, and satellite, etc. in the meteorological department (Abdelbaky et al., 2012). A variety of observation data are collected for meteorological services. Correspondingly, the accuracy and volume of the meteorological data are also increasing, the meteorological department not only owns these observation data, but also needs to generate and store many numerical forecast product data (Abdullahi et al., 2016). Generally, each type of meteorological data has different characteristics, including the frequency, quantity, type, and authenticity of the data (Alarabi et al., 2018). The explosive growth of meteorological big data also puts higher requirements on the real-time processing for such meteorological big data (Wang et al., 2013). The traditional meteorological processing architecture is far from meeting the real-time requirements (Abdelbaky et al., 2012). Therefore, it is extremely urgent to establish an efficient real-time processing platform for meteorological big data (Abdullahi et al., 2016; Chen et al., 2014).

While the meteorological big data puts higher requirements on data processing, the storage and sharing of the meteorological big data also receive much attention from the meteorological departments (Xu et al., 2018). At present, all the data in the meteorological departments are stored separately in their own independent physical servers, and the correlation between these servers is very small, but there is no unified integrated management for the physical servers (Hu et al., 2018; Ismail et al., 2017). So this traditional storage architecture cannot guarantee the real-time, reliability, availability, sharing, access efficiency and scalability of the meteorological big data (Lakshmanan and Humphrey, 2014), and it is imperative to establish the distributed management platform for the meteorological big data (Li et al., 2012).

As an effective and popular computing model and framework for big data parallel processing, the MapReduce framework is beneficial to forming a distributed and parallel computing cluster with thousands of nodes by using the existing physical servers in the meteorological department (Li et al., 2012, 2014). And the cluster-based high-performance parallel computing can fully meet the real-time requirements of the meteorological big data processing (Li and Shen, 2017). At the same time, for the storage requirements of the meteorological big data, the established platform employs the extensible system architecture, which can use multiple storage servers to share the storage load (Li et al., 2018; Ma et al., 2017). So the distributed storage architecture is easy to extend which can improve system reliability, availability, and access efficiency (Shuai, 2017).

Therefore, in response to the shortcomings and requirements of the current data processing and storage architecture of the meteorological department, we design and implement a *distributed data processing platform over meteorological big data using MapReduce*, which can obtain massive meteorological data from multiple data sources, and process the massive meteorological data in real-time based on MapReduce, and finally store the meteorological data in the distributed storage system to ensure the real-time, reliability, availability, sharing, access efficiency and scalability of the meteorological big data (Xin et al., 2017; Xu and Tang, 2016).

2 System analysis

2.1 Meteorological data sources analysis

At present, there are five main sources of data in the meteorological department: the FTP server, the file sharing server, the database server, the web-service interface and the various information websites.

- *FTP server:* The system platform connects one or more FTP servers through the LAN or WAN. The file update frequency may be different, and the file stored on the FTP server has download permission.
- *File sharing server:* Within the LAN, the meteorological department establish multiple remote data servers for sharing data. These file sharing servers mainly used to store various types of data files, and the update frequency of each file is uncertain.

- *Database server:* Some meteorological business systems usually save the collected data or processed product data to the database servers and provide data sharing for other hosts through permission settings.
- *Web-service interface:* The data provider publishes them in the form of web for sharing data.
- *Information websites:* Various types of websites are also the important data sources for the meteorological department.

2.2 Meteorological data analysis

According to the requirements of this platform, the meteorological data can be classified from the following two points:

- *Data type:* Different meteorological acquisition devices collect meteorological data for various purposes, and the recording format, data size, and acquisition frequency of each type of meteorological data are also different. At present, some meteorological data commonly found in meteorological departments mainly include the ground observation data, the high altitude data, the radar/satellite detection data, the numerical forecast products and the various service products.
- *Data structure:* According to the record structure of meteorological data, the meteorological data mainly includes structured data and unstructured data. Among them, the structured data mainly includes some data recorded in the text or data stored in the relational database, such as the automatic station data. But most meteorological data in the meteorological department are unstructured data, mainly including the radar/satellite detection data and some service products that contain video and sound.

2.3 System design objectives

In response to the shortcomings and requirements of the current data processing and storage architecture of the meteorological department, the designed distributed data processing Platform mainly needs to meet the following objectives:

- The platform can monitor the meteorological data of all data sources in real-time, and all changes of the meteorological data in the data source can be obtained in real-time.
- For all the massive meteorological data acquired in real-time, the MapReduce framework is used to process massive meteorological data in real-time to ensure the efficiency for data processing.
- After the completion of the processing, the meteorological data can be saved to the distributed storage system in real-time, facilitating the unified and centralised management of meteorological data.
- The platform not only ensures the real-time, reliability, sharing and access efficiency of the processing and storage for the meteorological data, but also has better scalability.

3 System design

3.1 System framework

According to the platform design goals described in Subsection 2.3, the overall framework of the platform can be divided into three layers: the data acquisition layer, the data processing layer, and the data storage layer.

- *The data acquisition layer:* According to the characteristics of various data sources and data in the current meteorological department, the corresponding real-time monitoring strategy is respectively formulated for real-time monitoring of meteorological data for all data sources. When data changes occur, the platform can obtain the information of the changed data in real-time and transmit the information to the *data processing layer* for processing.
- *The data processing layer:* After receiving the information of the changed data from the *data acquisition layer*, then according to the data type information, such as the meteorological automatic station data, the Micaps data or the radar detection data, etc. the MapReduce framework automatically calls the handler for this type of data, and process the meteorological data in parallel to guarantee real-time and efficient meteorological data processing.

  At present, the processing of various meteorological data mainly includes: data file judgment, data extraction, data cleaning, data calculation and data conversion. Different types of meteorological data, the specific process of data processing is also different.
- *The data storage layer:* When the *data processing layer* completes the real-time processing of the changed data, the final product data is transmitted to the *data storage layer*. In order to improve the efficient management and sharing of massive meteorological data, all meteorological data will be saved to the *distributed storage system* in real-time.

3.2 System function design

In Subsection 3.1, the three-layer architecture of the platform and the role of each layer are introduced. In this chapter, we will introduce the specific functions of each layer architecture.

3.2.1 The data acquisition layer

According to the types of data sources and data in the current meteorological department, combined with the real-time requirements of the data, we design the different data collection methods. In this platform, the *data acquisition layer* mainly adopts the following methods: real-time monitoring, regularly scan, database synchronisation, and web crawler.

- *Real-time monitoring:* Currently, many methods have been provided with real-time monitoring methods, such as the *FileSystemWatcher* method in the *.NET*

Framework, which can monitor the creation, deletion, modification, and renaming of files in the specified directory. When the file changes, the method can get the specific information of the changed file in real-time. The method is mainly applicable to monitoring data sources such as the file sharing server or the local server, and has high real-time performance.

- *Regularly scan:* Scanning the data sources per specified time to find out whether there is any new data or updated data during this time. The scanning frequency of this method is mainly set according to the update frequency of the data. For the data with a fixed updating frequency, the platform can set the frequency value according to the updating frequency of the data. Otherwise, the scanning frequency of the platform is mainly determined by the real-time requirements of the data and the server workload. The higher the scanning frequency, the higher the real-time performance, but the greater the load on the server. Therefore, under the premise of ensuring the normal operation of the server, the higher the real-time requirement, the higher the scanning frequency, and vice versa. In this platform, this method is mainly applied to the monitoring of the FTP servers.
- *Database synchronisation:* Based on the current database synchronisation technology, such as setting the publish and subscribe modes for the master and slave databases respectively, or writing *triggers* to the target database to achieve real-time data synchronisation in the target database. The method is mainly applied to synchronising the databases of various meteorological service systems in the current meteorological department.
- *Web crawler:* For the current business in meteorological department, a lot of information comes from various websites, such as China Weather Network. In such information websites, some of them provide the *web-service interface*, and the platform can obtain data through these interfaces, but for the websites that do not provide the data interfaces, the platform mainly adopts *web crawler* method, according to the certain rules, automatically getting the information from the target websites.

3.2.2 The data processing layer

As the core module of the platform, the function of the *data processing layer* is mainly to process all kinds of meteorological data collected by the *data acquisition layer* in real-time. The processing of meteorological data mainly includes the following types: the file judgment, the data extraction, the data cleaning, the data calculation and the data conversion:

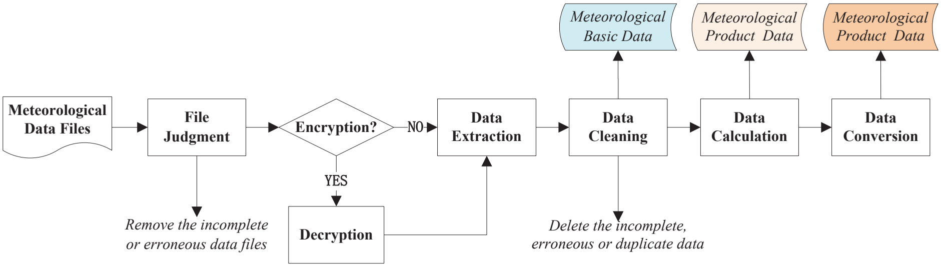
- *File judgment:* For all kinds of the meteorological data collected in real-time by the *data acquisition layer*, not all files are the required data files. At the same time, there are also some incomplete data files or incorrect data files in the collected data files. Therefore, the platform needs to remove these files.
- *Data extraction:* According to the data format of various meteorological documents, the required meteorological data are extracted. At present, the recording format of most meteorological data is relatively simple. Only some data

files use encryption technology, so it needs to be decoded before the required data can be extracted.

- *Data cleaning:* After data extraction, there are a large amount of the incomplete data, the erroneous data, and the repeated data. The platform handles the lost data, the cross-border data, the inconsistent data, and the duplicate data based on the accuracy, completeness, consistency, uniqueness, timeliness and effectiveness of the data.
- *Data calculation:* Most of the data collected by the *data acquisition layer* are the meteorological basic data, such as the ground observation data. The value of these data can be used directly is limited, so the further calculations are needed for the basic data, such as statistics and analysis, etc. The generated product data can only play the value of these basic data. For example, through the observation data of the meteorological automatic station, the daily maximum temperature and the lowest temperature of each meteorological automatic station are separately counted.
- *Data conversion:* According to the different data application requirements, even the same meteorological data may need to be converted into the data in another format. For example, for some data with space-time characteristics, the platform needs to interpolate these data, generate contours/equivalents, smoothing, and layer overlays, and finally generate spatial GIS data with spatiotemporal characteristics.

So the data processing flow chart of the *data processing layer* is as shown in Figure 1.

Figure 1 The data processing flow chart of the data processing layer (see online version for colours)


Meteorological Basic Data
Meteorological Product Data
Meteorological Product Data
Meteorological Data Files
File Judgment
Encryption?
NO
YES
Data Extraction
Data Cleaning
Data Calculation
Data Conversion
Remove the incomplete or erroneous data files
Decryption
Delete the incomplete, erroneous or duplicate data

In short, the types of meteorological data are different, the respective reading formats, data calculation requirements, and finally generated product data formats are also different.

3.2.3 The data storage layer

After the collected data is processed in the *data processing layer*, the data will be saved to the *distributed storage system* in real-time to realise unified centralised management and sharing of meteorological data.

4 System architecture

In order to meet the need of meteorological department for efficient processing and storage of massive meteorological data and achieve the sharing of massive meteorological data, we employ the MapReduce paradigm to specify our designed platform. The platform can effectively solve the shortcomings of the *meteorological big data* in real-time processing and storage in the traditional way.

At present, the traditional meteorological data processing architecture mainly uses high-performance servers for data processing and storage. However, with the explosive growth of meteorological data, due to the limitation of hardware resources, dealing with the massive meteorological big data on a single server has gradually become incompetent.

With the maturity of big data processing and storage technology, this paper proposes to build all the servers in the current meteorological department into a cluster, and each server is regarded as a node in the cluster. Then, the MapReduce framework is introduced to perform distributed processing on the massive meteorological big data in the *data processing layer*. The processing tasks executed on a single server in the traditional architecture are distributed to the nodes where the data is located, and each node separately performs the processing tasks of the data on the respective nodes, which is the *map phase*. When each node completes its own processing tasks, it produce its own intermediate results. The *reduce phase* collects the intermediate results and summarises them into the final result.

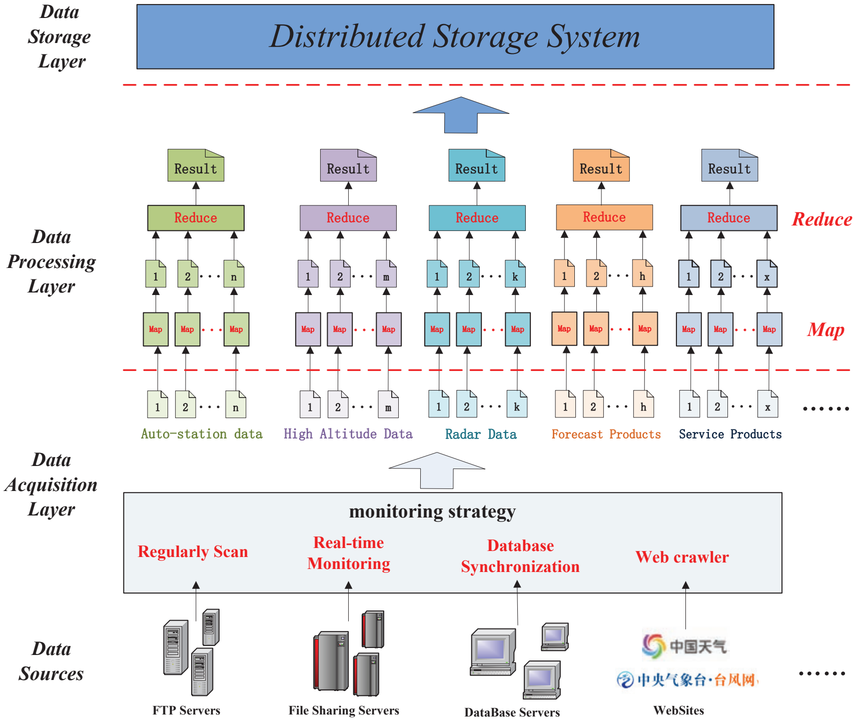
With the maturity of big data processing and storage technology, all the servers in the current meteorological department are grouped into a cluster which can perform various tasks in parallel to achieve higher computing speeds. Besides, each server in the cluster is regarded as a node and backs up each other. Then, the MapReduce framework is introduced to perform distributed processing on the massive meteorological big data in the *data processing layer*. The tasks executed on a single server in the traditional architecture are distributed to the multiple servers where the data is located, and each server separately performs the processing tasks on the respective nodes, which is the *map phase*. Each node produce an intermediate result after completing its own tasks. and all intermediate results are combined into the final result in the *reduce phase*.

The *distributed storage* is used in the *data storage layer*. Distributing massive meteorological big data to multiple storage nodes helps to alleviate the storage and sharing pressure of a single storage server in the traditional storage architecture. Besides, it also ensures the security of meteorological data. Finally, the distributed storage system uniformly manages and allocates the resources of nodes in the cluster, and provides the user with the access interface of massive meteorological data.

Compared with the traditional meteorological data processing-storage architecture, the *distributed architecture based on MapReduce* can effectively alleviate the pressure of the single server and improve the processing and storage efficiency of meteorological big data. Besides, when the computing and storage performance of the cluster is insufficient, the meteorological department can also extend the performance of the cluster by adding nodes. And when a node fails, its computing tasks can be migrated to other nodes for execution, so the MapReduce architecture also has good scalability and high fault tolerance.

The architecture of the *distributed data processing platform over meteorological big data using MapReduce* is shown in Figure 2.

Figure 2 The platform architecture (see online version for colours)



5 The application examples of platform

According to the architecture of the *distributed data processing platform over meteorological big data using MapReduce*, the Shanghai Jiading Meteorological Service designed and implemented the *cloud platform for comprehensive processing of meteorological disaster information*.

The cloud platform acquires various types of meteorological data from various data sources in real-time at the *data acquisition layer*. For example, the platform obtains the observation data of all automatic stations around Shanghai and Shanghai from the FTP servers or the local servers, obtains the massive historical meteorological disaster data from multiple database servers, and obtains the national weather data from websites such as the *China Weather Network*. All data is processed in parallel based on the MapReduce framework at the *data processing layer* of platform. For example, the data of all meteorological automatic stations are numerically counted and interpolated according to each meteorological element to generate the isosurface. Then the platform analyses the various meteorological element data and the massive historical meteorological disaster data through the *meteorological disaster prediction model* to generate the *meteorological disaster risk prediction maps*. Finally, all data collected by the *data acquisition layer* and the product data processed by the *data processing layer* are stored to the *distributed storage system*.

Figure 3 Risk prediction map of rainstorm disaster (see online version for colours)

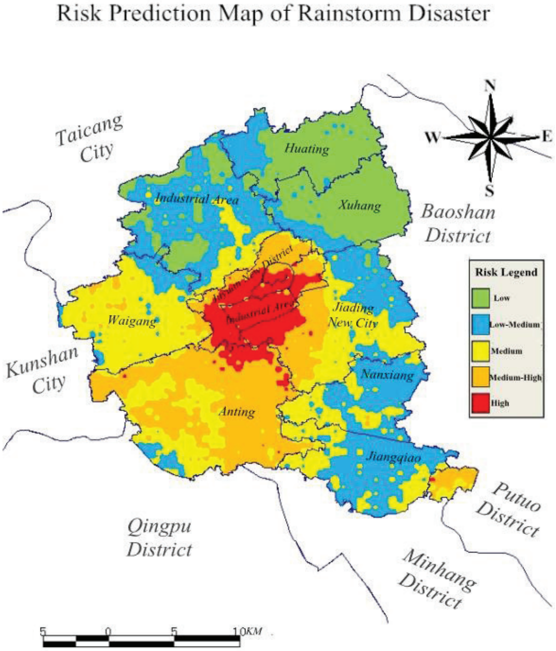
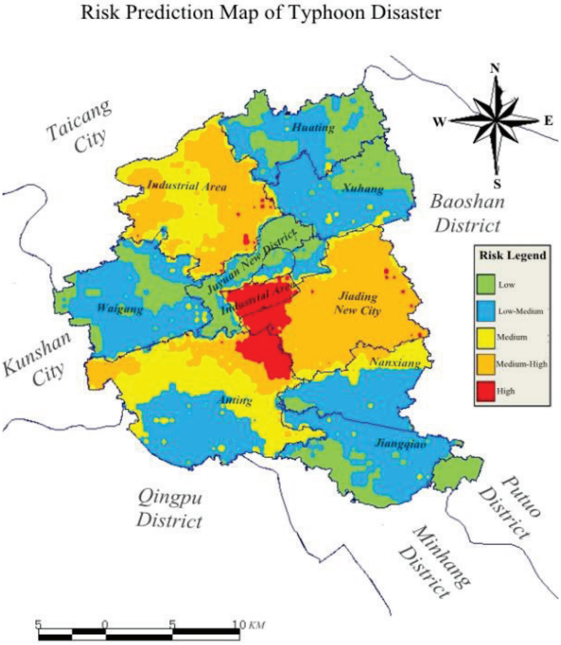


Figure 4 Risk prediction map of typhoon disaster (see online version for colours)



In the *data processing layer* of the platform, combined with various meteorological data and massive historical meteorological disaster data, through the calculation and analysis of the meteorological disaster prediction model, the platform generates the *risk prediction maps* for various meteorological disasters, such as the *risk prediction map of rainstorm disaster* and the *risk prediction map of typhoon disaster*, as shown in Figures 3 and 4.

6 Conclusions and future work

This paper focuses on the distributed processing and storage of massive meteorological data in the meteorological department. In view of the challenge, the *distributed data processing platform over meteorological big data using MapReduce* is proposed. This platform can monitor all major data sources of current meteorological department in real-time and obtains massive meteorological data. Combined with the current mature big data processing and storage technology, the MapReduce framework is used for distributed parallel processing of massive meteorological data, and finally all kinds of meteorological data are stored in the distributed storage system. The platform not only improves the processing and storage efficiency of meteorological big data, but also has good scalability and fault tolerance.

Finally, as the instantiation in Shanghai Jiading Meteorological Service, the *cloud platform for comprehensive processing of meteorological disaster information* fully confirms the feasibility and the practical applicability of the *distributed data processing platform over meteorological big data using MapReduce*.

However, the shortcomings of this platform are also obvious, especially the processing of complex meteorological data is not smart enough. At the same time, in the storage of meteorological data, the placement optimisation of massive meteorological data has not been considered to reduce the average data acquisition time of each meteorological application in the cluster. So in future work, we will continue to optimise the storage and computing architecture of meteorological data.

References

Abdelbaky, M., Kim, H., Rodero, I. and Parashar, M. (2012) 'Accelerating MapReduce analytics using cometcloud', in *IEEE Fifth International Conference on Cloud Computing*.

Abdullahi, A.U., Ahmad, R. and Zakaria, N.M. (2016) 'Big data: performance profiling of meteorological and oceanographic data on hive', in *International Conference on Computer & Information Sciences*.

Alarabi, L., Mokbel, M.F. and Musleh, M. (2018) 'St-Hadoop: a MapReduce framework for spatio-temporal data', *GeoInformatica*, Vol. 22, No. 4, pp.785–813.

Chen, D., Zeng, L., Liang, Z. and Xiao, W. (2014) 'HBase-based distributed storage system for meteorological gound minute data', *Journal of Computer Applications*, Vol. 34, No. 9, pp.2617–2621.

Hu, C., Li, W., Cheng, X., Yu, J., Wang, S. and Bie, R. (2018) 'A secure and verifiable access control scheme for big data storage in clouds', *IEEE Transactions on Big Data*, Vol. 4, No. 3, pp.341–355.

Ismail, K.A., Majid, M.A., Zain, J.M. and Bakar, N.A.A. (2017) 'Big data prediction framework for weather temperature based on MapReduce algorithm', in *Open Systems*.

Lakshmanan, V. and Humphrey, T.W. (2014) 'A MapReduce technique to mosaic continental-scale weather radar data in real-time', *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing*, Vol. 7, No. 2, pp.721–732.

Li, Z. and Shen, H. (2017) 'Measuring scale-up and scale-out Hadoop with remote and local file systems and selecting the best platform', *IEEE Transactions on Parallel & Distributed Systems*, No. 99, p.1.

Li, B., Mazur, E., Diao, Y., Mcgregor, A. and Shenoy, P. (2012) 'Scalla: a platform for scalable one-pass analytics using MapReduce', *Acm Transactions on Database Systems*, Vol. 37, No. 4, pp.1–43.

Li, J., Chen, X., Li, M., Li, J., Lee, P.P.C. and Lou, W. (2014) 'Secure deduplication with efficient and reliable convergent key management', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25, No. 6, pp.1615–1625.

Li, J., Wang, J., Lyu, B., Wu, J. and Yang, X. (2018) 'An improved algorithm for optimizing MapReduce based on locality and overlapping', *Tsinghua Science and Technology*, Vol. 23, No. 6, pp.112–121.

Ma, X., Fan, X., Liu, J., Jiang, H. and Kai, P. (2017) 'Vlocality: revisiting data locality for MapReduce in virtualized clouds', *IEEE Network*, Vol. 31, No. 1, pp.28–35.

Shuai, Z. (2017) 'Application-aware network design for Hadoop MapReduce optimization using software-defined networking', *IEEE Transactions on Network & Service Management*, No. 99, p.1.

Wang, L., Tao, J., Ranjan, R., Marten, H., Streit, A., Chen, J. and Chen, D. (2013) 'G-Hadoop: MapReduce across distributed data centers for data-intensive; computing', *Future Generation Computer Systems*, Vol. 29, No. 3, pp.739–750.

Xin, L., Zhao, D., Liang, X., Zhang, W., Yin, J. and Chen, X. (2017) 'A distributed video management cloud platform using Hadoop', *IEEE Access*, Vol. 3, No. 1, pp.2637–2643.

Xu, X. and Tang, M. (2016) 'A new approach to the cloud-based heterogeneous MapReduce placement problem', *IEEE Transactions on Services Computing*, No. 99, pp.862–871.

Xu, X., Liu, Q., Luo, Y., Peng, K., Zhang, X., Meng, S. and Qi, L. (2018) 'A computation offloading method over big data for IoT-enabled cloud-edge computing', *Future Generation Computer Systems*, Vol. 95, No. 2, pp.522–533.