Int. J. Advance Soft Compu. Appl, Vol. 7, No. 3, November 2015
ISSN 2074-8523

Enhancing Security and Privacy Protection for MapReduce Processing: The Initial Simulation Work Flow

Adilah Sabtu, Nurulhuda Firdaus Mohd Azmi, and
Siti Sophiayati Yuhaniz

Advanced Informatics School (UTM AIS)
Universiti Teknologi Malaysia,
Jln Sultan Yahya Petra, 54100 Kuala Lumpur
e mail: adilah.sabtu@gmail.com, huda@utm.my,
sophia@utm.my

Abstract

MapReduce programming model allows the processing of massive amount of data in parallel through clustering across a distributed system. The tasks for MapReduce have been categorized into areas which are data management and storage, data analytics, on line processing and security and privacy protection. For sensitive data uploaded by users, it must be protected from any unauthorized access to ensure the integrity, authenticity and privacy of the data. It is important that, data at rest, data in transit and nodes is managed securely by ad-dressing the elements of data security and privacy protection which are auditing, access control and privacy. The purpose of this study is to enhance the prominence of security and privacy protection for MapReduce model. An existing study is more specifically tailored to structure based requirements. Whilst, there is a need to continue finding solutions for MapReduce in better handling big data security and privacy protection concerning the unstructured data. This paper presents the initial workflow of the simulation set up of MapReduce processing using the Hadoop platform to demonstrate an enhancement for security and privacy protection access control by implementing Whitelist to control access in MapReduce processing.

Keywords: *Big Data Processing, MapReduce Programming Model, Security and Privacy Protection, Hadoop Platform.*

1 Introduction

Generally, big data is a term used to describe high volume that extended into the Terabyte and Petabyte range, high velocity and highly complex dataset consisting of structured, semi-structured and unstructured data which require new ways, technologies, techniques and mindsets to scale, harness and trans-form the data into useful, insightful, real-time and comprehensive information for analysis and decision making. The type of big data have been further classified as traditional enterprise data, machine generated/sensor data and social data. A key challenge for a big data is dealing with the growth of information since unstructured data are hard to analyze and expressed in a united model. The characteristics of the usual data after evolving into a superlative type have rendered the relational data model commonly used in a relational database management system today short in the capacity to tap and handle big data potential effectively. Contrary to the customary models that work with structured data to achieve accuracy, big data consisted of semi-structured and free-form data invites prediction. Big data processing is basically characterized by the scalability for large scale data-intensive computing and fault tolerance, adaptation in diverse environments, support schema-free format and real-time large batch processing. MapReduce programming model allows the processing of massive amount of data in parallel through clustering across a distributed system. The tasks for MapReduce have been categorized into areas which are data management and storage, data analytics, online processing and security and privacy protection.

For MapReduce processing, data are replicated across multiple nodes with high-speed and parallel processing nodes running on very large sets of data. Sensitive data uploaded by user must be protected from any unauthorized access to ensure the integrity, authenticity and privacy of the data. It is important that data at rest, data in transit and nodes are managed securely by addressing the elements of data security and privacy protection task which are identified as auditing, access control and privacy or other possible elements. In big data community, MapReduce has been considered as a highly efficient tool for processing massive datasets mainly because of its elastic scalability and fault tolerance behavior [7, 9, 19]. However, security and privacy concern for data that negotiated the clusters are still largely dealt with using tradi-tional approaches which need new security requirements for big data to be effective [12, 25]. MapReduce security and privacy protection task elements could use as basis of selecting security and privacy protection options. Expec-tation of the simulation of access control element for MapReduce processing may contribute to better understanding of security and privacy protection elements enhancement of a MapReduce model.

The purpose of this study is to enhance the prominence of security and privacy protection for MapReduce model. An existing study is more specifically tailored to structure based requirements. Whilst, there is a need to continue finding solutions for MapReduce in better handling big data security and privacy protection concerning the unstructured data. This paper presents the initial workflow of the simulation set up of MapReduce processing using the Hadoop platform to demonstrate an enhancement for security and privacy protection access control by implementing Whitelist to control access in MapReduce processing.

2 Big Data Programming Models

Ideally, Big Data programming model should be able to provide computing resources imposed by massive datasets requirements composed of all kinds of data. This means that a programming model should support both distributed data management and processing, high-level language such as SQL, iterative algorithms, iterative ad-hoc data exploration and stream processing for structured data and at the same time support high scalability, fault-tolerance, simplicity, schema-free, ignore data storage system and parallel data processing across a large cluster of nodes for the unstructured data [10,21]. Fig. 1 shows the general model layer of Big Data. The existing Big Data programming models are Relational Database Management System (RDBMS) [21], NoSQL ((Not only) SQL) [9] and MapReduce [10,14]. Section 2.1 will describe further about MapReduce programming model.

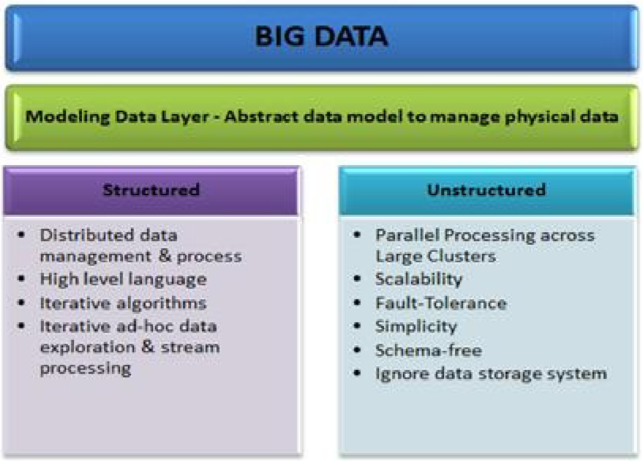
BIG DATA
Modeling Data Layer - Abstract data model to manage physical data
Structured
• Distributed data management & process
• High level language
• Iterative algorithms
• Iterative ad-hoc data exploration & stream processing
Unstructured
• Parallel Processing across Large Clusters
• Scalability
• Fault-Tolerance
• Simplicity
• Schema-free
• Ignore data storage system

Fig. 1: Framework General Model Layer of Big Data [2,20]

2.1 MapReduce Programming Model

MapReduce inspired by Google is a functional programming of two structural transformation functions, Map and Reduce, composed together in practice to provide a programming interface for implementing algorithms and to perform wide variety of computations. MapReduce is claimed as parallel programming model designed based on divide-and-conquer concept and follows a master/slave paradigm [24]. Furthermore, Qin et. al [21] described MapReduce as a general execution engine that ignores storage layouts and data schema, and the runtime system automatically parallelizes computation across a large cluster of machines, handles failure and manages disk and network efficiency. While, Zhang et. al [25] described MapReduce as a scalable and fault-tolerant data processing framework that is capable of processing huge volume of data in parallel with many low-end commodity computers. Work by Fadika et. al. [5] proposed that MapReduce model is anchored around 3 central princi-ples, namely, the data management, synchronization/ parallelization abstraction and fault-tolerance. These studies shows that MapReduce allows the processing of massive amount of data in parallel through clustering across a distributed system which applies both to structured and unstructured data. Furthermore, the popularity of MapReduce can be accredited to its high scalability, fault-tolerance, simplicity and independence from the programming language or the data storage system [14].

In MapReduce operations [19,23] (as depicted in Fig 2), Map will accept input of data, a master controls and distributes the data to worker nodes with the tasks of assigning keys and values in pairs (key, value) and stores the results as list of key/value sets (key, ? list of values ?), master handles the distribution of the clusters (collection of nodes), monitors and tracks the process. Taking advantage of the locality of data, a Shuffle step is inserted between Map and Reduce where it will aggregate and merge the key/value with matching pairs. The distribution of data takes place at worker node based on the output keys from Map and all data belonging to the same key are located on the same node. In a more diverse distributed file system, similar sub-step coined as Combine may be added preceding the Shuffle, in order to ease up the merging tasks for different subtasks or file systems.

Reduce then applies its operations using similar master/slave approach; gets the keys and list of values and sums up the values, resulting in the final (key, value) output. Splitting of data among clusters allows for redundancy and fault-tolerance in MapReduce while processing data in parallel means the capacity to process data is faster which works in favor of big data characteristics. Also, as businesses expand, MapReduce sub-tasks may be needed to fulfill some business requirements and avoid from having to tackle the whole complex system.

3 MapReduce in Big Data Challenges

Grolinger et. al. [10], conducted a research of MapReduce by focusing on identifying MapReduce challenges in big data. The study categorized the

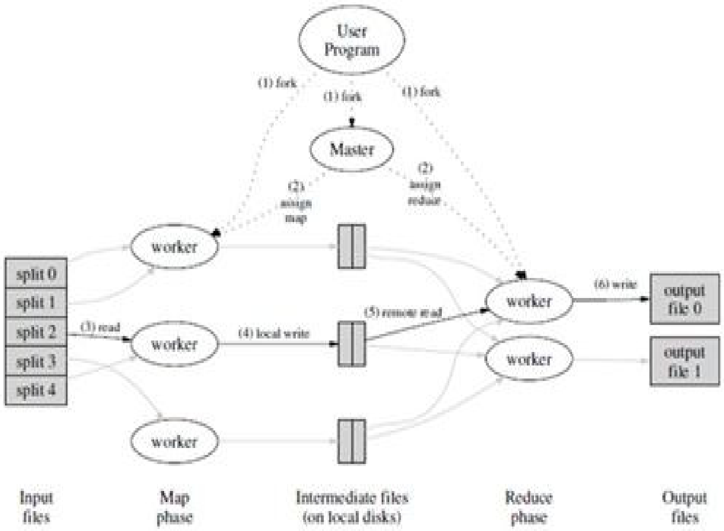

Fig. 2: Model of Googles MapReduce Flow [10,19,23]

challenges based on task types namely, data storage, data analytics, online processing and security and privacy. The strengths of MapReduce in data stor-age are its schema-free and index-free attributes which provides flexibility and scalability when handling semi-structured and unstructured data. However, the lack of index could also lower the performance of MapReduce compared to relational databases.

In addition, Grolinger et. al. [10] identified several weaknesses that can be improved such as lack of standardized SQL-like query language, limited optimization jobs and integration issues among existing models or systems. For big data analytics, MapReduce faced the challenge of implementing machine language algorithm despite its parallel aspect. Low-latency processing in on-line processing was also identified as a weakness for MapReduce. Security and privacy challenges described in MapReduce pertain to audit, access control and privacy. Fig. 3 below shows the relationships between MapReduce tasks and its impact on big data challenges. Tekiner & Keane [20] outlined big data challenges as the need to process data of immense volume and scale, the analytics used to extricate value from variety and heterogeneity of data sources, the speed and timeliness of information requirements, requirement of new targeted services, products, solutions and applications, data presentation, usability and interpretation and finally data privacy, error handling and security challenges. In contrast, study by Garcia [7] highlighted mainly on MapReduce algorithm design strategies and limitations. It discussed about object word sentiment analysis and stressed that the intermediate key produced by Map function as a critical bridge between Map and Reduce functions and on load balancing. The research further implied that MapReduce is useful for big data tasks that do not involve excessive iterative refinement and processing coordination.

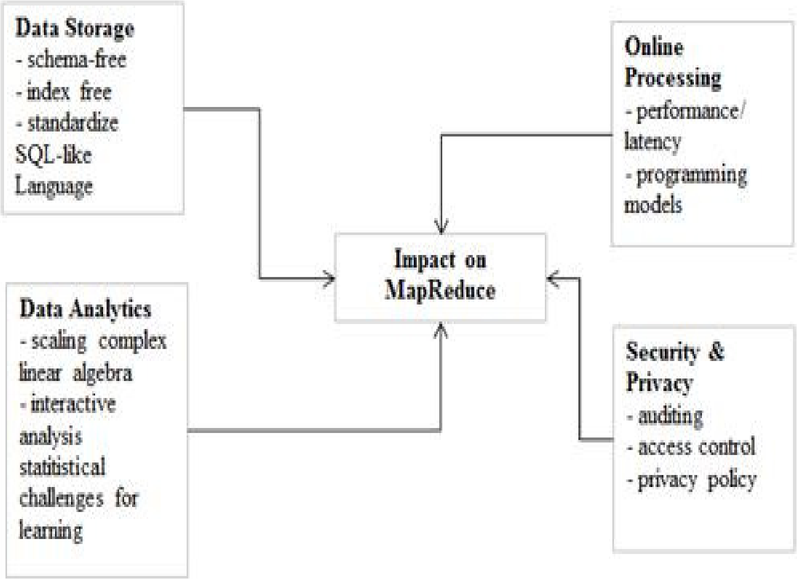

Fig. 3: MapReduce Tasks for Handling Big Data Challenges, [10]

Pandey & Tokekar [19] presented big data challenges as management and storage, calculation and examination and big data privacy protection. The research further detailed MapReduce contributions by making a comparison study between MapReduce and Parallel DBMS on basis of various parameters, MapReduce performance parameters on the basis of its architectural de-sign and the optimization of various parameters based on programming model, storage independent and run time scheduling. The strengths of MapReduce were identified as fault tolerance, storage system independence, exibility and simplicity. According to the study in [18,22] and [24], security and privacy requires the management of dynamic data mining without exposing the sensitive information of individuals. Current technologies of implementing security and privacy protection are still tailored for static data. There was a limited elaboration on how MapReduce performed against data privacy protection issue dynamically.

4 Related Works on Security and Privacy Protection in MapReduce

Security measures become difficult to catch up with the acceleration rate of big data. In a survey recently released by Information Week 2014 indicated that even across the internet sphere, security of big data is at best shaky. For some enterprises, the quest for data takes precedence over security. Hu et. al. [11] in their study mentioned that big data platform should find a good balance between enforcing data access control and facilitating data processing.

An example of study on security and privacy protection in MapReduce can be directed to Xu et. al. [24] whereby they conducted a research of cloud computing based system for cyber security management. Apache Hadoop was used as the open-source framework and used to analyze cyber security awareness. Hadoop provide useful functions such as Network and Capture to capture Network Traffic, Netflow Collection which collect and store Netflow data into binary files and Traffic Information Storage which is used to read Netflow data from files and dump the in store as plain-text files. The cloud-based system introduced system architecture, data storage module and task scheduling module before the implementation. The efficiency of overall cyber security situation awareness was improved by implementing MapReduce in the cloud infrastructure through fast retrieval and processing. The research focused on analyzing network traffic for abnormal behaviors or patterns. Performance evaluation of this system especially considered two scenarios, both based on MapReduce platform which were ranking for scanning end-user devices traffic and aggregation used to consolidate analyzed result and conduct statistical analysis for intrusion detection.

In order to classify the related works on security and privacy protection for MapReduce, we classify the studies based on the security and privacy elements as described in Table 1.

The security and privacy and protection elements mentioned in Table 1 can be further described as follows [23]:

1. Auditing is related to the creation of audit trails that tracks events that occurred in data stores.
2. Access control mechanism allows or restricts access to multiple storage locations or devices where semantic understanding should influence access control decision process. There are basically 3 types of access controls mechanisms that provide different level of protection:

    Policy-based Access Control - The most popular policy-based access control method is the Access Control List (ACL) which is a list of authorized users with specific permissions for reading, writing and execution [1]. The list is automatically referenced each time a service is requested.

Table 1: Security and Privacy Protection Elements

Element	Solution Approach
Accountability and Auditing	• Trusted 3rd party monitoring • Security analytics
Access Control	Optimized access control approach with semantic understanding
Privacy	• Policy enforcement with security to prevent information leakage • Refined anonymization
Parameter	Optimized processing within tolerable time

Discretionary Access Control (DAC) - DAC lets owner of a file or physical object permission to allow or deny access to users based on assigned roles or content-dependent access. Role-based Access Control (RBAC) is based on the role the user is assigned and least privilege concept which means access is denied unless required to complete a job. It simplifies the management of access rights. Content-dependent Access Control lets users to access only the contents their privileges allow.

Mandatory Access Control (MAC) - MAC is an access policy supported for systems that process especially sensitive data where labels must be assigned to all subjects to determine who to access what information and what request in the systems. Indrajit et. al. [12] stated that MAC systems check permissions on every operation and transitively enforce access restrictions and enforce access rules specified by the system administrator at all times, without user override.

3. Privacy pertains to protection on the data by applying privacy rules such as what category of the data, entities that can access the data, purposes and conditions to use the data [14].

4. Parameters basically should perform processing that enhances data security.

The elements identified in Table 1 will be as the indicator for the identified existing works on security and privacy protection in MapReduce model. The identified related works is summarize in Table 2.

Table 2: Related Works on Security and Privacy Protection in MapReduce

Research	Security and Privacy Protection Element				Platform
	Auditing	Access Control	Privacy	Parameter	
Indrajit et. al. [12]	Untrusted code audit not needed	Mandatory Access Control	Differential Privacy	Various parameters to different participants	Hadoop
Zhang et.al. [25]	-	-	Sub-tree anonymization	Top Down Specialization, Bottom Up Generalization	U-Cloud
Dyer & Zhang [4]		Authenticate service	-	-	Virtual Machine in public cloud

5 The Initial Simulation Work Flow

This study will be based on causal relationship theory design and carry out an experimental simulation of the security and privacy protection enhancement for MapReduce. Simulation is a controlled computer-based experiment with logic models and uses a model-building technique to determine the effects of changes. In order to carry out the simulation, the important elements of security and privacy protection task for MapReduce model should be clearly defined and determined in order to decide and design the simulation with enhanced security and privacy protection element for MapReduce using Hadoop platform. The following Fig. 4 shows an initial workflow of the simulation. The study will use Hadoop architecture in Hortonworks platform widely used for big data analytics which has its environment built with MapReduce and HDFS as its core.

To describe the workflow, firstly, the scripts containing Whitelists of specifics is run into series of MapReduce jobs on the Apache Hadoop cluster. Next, the available datasets selections is used in the framework. Finally, the results will

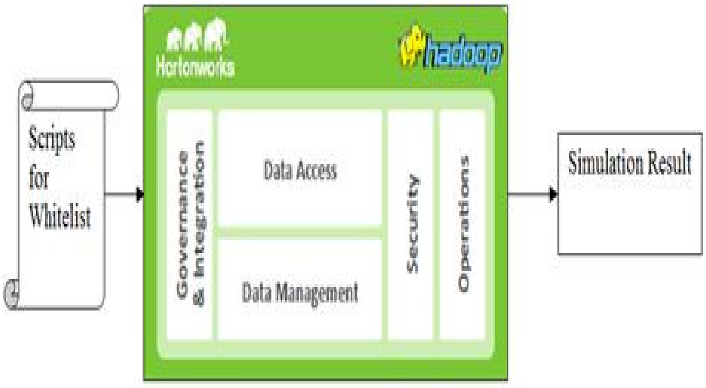

Fig. 4 Simulation Work Flow

be download from file browser for analysis. This simulation is planned to cre-ate two Whitelist scripts that control accesses into the system. It is a form of access control and safety precaution to grant files \express pass" into the system and later differentiate it from untrusted source. Whitelist can reduce false positives Type 1 error. The Whitelists scripts are described as follows:

Whitelist 1 contains trusted files added as mandatory access control that ensure a list of trusted files allowed to access into the MapReduce framework.

Whitelist 2 contains trusted domains added as mandatory access control that will restrict mappers residing in untrusted domains from access.

The simulation is expected to observe and analyze the following results:

All trusted files and domains listed in the Whitelists are allowed access into the nodes for processing.

Any files not listed in the Whitelists should leak pass the nodes.

Process execution with Whitelist scripts should be within reasonable time comparatively with the process without Whitelist.

The test-bed simulation which will be develop in this study is not in a real-time and distributed environment and is expected to expose to a control setup for investigation purposes. Comparative trial will be carried out with different sets of input in order to achieve the e effectiveness of MapReduce processing for security and privacy protection.

6 Conclusion

The ecosystem for MapReduce which is synonymous with processing unstructured data is still developing. Security and privacy protection aspects of MapReduce should be continually improved in order to ensure data integrity in big data clusters. This study attempts to simulate enhancement of security and privacy access control element for MapReduce processing with Whitelists on a Hadoop platform. In order to carry out the simulation, the important elements of security and privacy protection task for MapReduce model should be clearly de ned and determined. The security and the privacy elements have been discussed in this paper which consists of auditing, access control, privacy and parameter. At the end of the paper, we have discussed the expected input and output for the planned simulation. Our next focus is to test and analyze the e effectiveness of the simulations aiming to enhance the promi-nence of security and privacy protection task elements for a MapReduce model.

ACKNOWLEDGEMENTS
The authors would like to thank the Ministry of Higher Education of Malaysia (MOHE) for the funded given in Fundamental Research Grant (FRGS) and Universiti Teknologi Malaysia (UTM) for the support of the resources. This work is currently funded by a FRGS Grant Vot Number:4F266.

References

[1] Adamson, C.L. & Wood, A.G (2010). DFBIdb: A Software Package for Neuroimaging Data Management. Neuroinform. 8. 273 284

[2] John G. Daugman. 1985. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two dimensional visual cortical lters. J. Optical Soc. Amer. A: Optics, Image Science, Vision 2, 7 (1985), 11601169.

[3] Chen, P. P.-S. (1976). The Entity-relationship Model & Mdash; Toward a Unified View of Data. ACM Trans Database Syst. 1(1), 936.

[4] Dyer, J. and Zhang, N. (2013). Security Issues Relating to Inadequate Authentication in MapReduce applications. IEEE. July 2013. 281 288.

[5] Fadika, Z., Dede, E., Govindaraju, M. and Ramakrishnan, L. (2011). Benchmarking MapReduce Implementations for Application Usage Scenarios. 12th IEEE/ACM International Conference. 21 23 September 2011. Lyon, 90 97.

[6] Gantz, J and Reinsel, D. (2012). The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. IDC Anal. Future.

[7] Garcia, C. (2013). Demystifying MapReduce. Procedia Computer Science. 20, 484 489.

[8] Goss, R.G. and Veeramuthu. K. (2013). Heading Towards Big Data Building a Better Data Warehouse for More Data, More Speed, and More Users. 2013 24th Annual SEMI. 2013. 220225.

[9] Grolinger et al. (2013). Data Management In Cloud Environments: NoSQL and NewSQL data stores. Journal of Cloud Computing: Ad-vances, Systems and Applications. 2(22).

[10] Grolinger, K., Hayes, M., Higashino, W.A., LHeureux, A., Allison, D.S. and Capretz, M.A.M. (2014). Challenges for MapReduce in Big Data. 2014 IEEE World Congress on Services (SERVICES). 2014. 182189.

[11] Hu, H., Wen, Y., Chua, T-S., Li, X. (2014). Toward Scalable Systems for Big Data Analytics: A Technology Tutorial. IEEE Access. 2, 652 687.

[12] Indrajit, R., Srinath, T.V.S., Ann, K., Vitaly, S., Emmett, W. (2010). Airavat: Security and Privacy for MapReduce. 2010 Proceedings of the 7th Usenix Symposium on Networked Systems Design and Implementation. 2010. San Jose, USA.

[13] Jinbao, Z. and Wang, A. (2012). Data Modeling for Big Data. Unpublished. CA Technologies.

[14] Kassim, H., Hung, T. and Li, X. (2012). Data Value Chain as a Service Framework: for Enabling Data Handling, Data Security and Data Anal-ysis in the Cloud. 2012 IEEE 18th International Conference on Parallel and Distribution Systems. December 2012. 804 809.

[15] Kubo, R., Fukumoto, Y. and Onizuka, M. (2012). Efficient Large-Scale Data Analysis Using MapReduce. NTT Technical Review. 10(12), 1 6.

[16] Lehtinen, R., Russell, D. & Gangemi Sr, G.T. (2006). Computer Security Basics. (2nd). Sebastopol: OReilly.

[17] Liu, Z., Yang, P. and Zhang, L. (2013). A Sketch of Big Data Technologies. 2013 Seventh International Conference on Internet Computing for Engineering and Science (ICICSE). 2013. 2629.

[18] Lu, T., Guo, X., Xu, B., Zhao, L., Peng, Y. and Yan, H. (2013). Next Big Thing in Big Data: The Security of the ICT Supply Chain. 2013 International Conference on Social Computing (SocialCom). September 2013. 1066 1073.

[19] S. and Tokekar, V. (2014). Prominence of MapReduce in Big Data Processing. 2014 Fourth International Conference on Communication Systems and Network Technologies (CSNT). April 2014. 555 560.

[20] Tekiner, F. and Keane, J.A. Big Data Framework (2013). 2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC). 2013. 14941499.

[21] Qin, X., Wang, H., Li, F., Zhao, B., Cao, Y., Li, C., Chen, H., Zhou, X., Du, X. and Wang, S. (2012). Beyond Simple Integration of RDBMS and MapReduce Paving the Way toward a Unified System for Big Data Analytics: Vision and Progress. 2012 Second International Conference on Cloud and Green Computing (CGC). November 2012. 716 725.

[22] Wei, W., Du, J., Yu, T., and Gu, X. (2009). SecureMR: A Service Integrity Assurance Framework for MapReduce. Proceedings of the 2009 Annual Computer Security Applications Conference. 2009. Washington DC: IEEE. 73 82.

[23] Xiao, X., Tang, J., Chen, Z., Xu, J. and Wang, C. (2015). A cross-job framework for MapReduce scheduling. 2014 IEEE International Conference on Big Data, IEEE Big Data 2014. 2015. 135 140.

[24] Xu, G., Yu, W., Chen, Z., Zhang, H., Moulema, P., Fu, X. and Lu, C. (2015). A Cloud Computing Based System for Cyber Security Management. International Journal of Parallel, Emergent and Distributed Systems. 30(1), 29 45.

[25] Zhang, X., Liu, C., Nepal, S., Yang, C., Dou, W. and Chen, J. (2013). Combining Top-Down and Bottom-Up: Scalable Sub-tree Anonymization over Big Data Using MapReduce on Cloud. 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). July 2013. 501 508.