SAJET

South Asian Journal of Engineering and Technology Vol.2, No.17 (2016) 51–57

ISSN No: 2454-9614

Privacy ensured High scalable Data Analysis under Mixed Cloud Architecture

G.K. Jhanani, M. Kavya*, J. Kiruba, R. Rajeshwari, S. Viajyanand

Department of Computer Science and Engineeering, Shree Venkateshwara Hi-Tech Engineering College , Gobi, Tamilnadu, India.

*Corresponding Author: G.K. Jhanani

E-mail: jhanani15k@gmail.com

Received: 09/11/2015, Revised: 15/12/2015 and Accepted: 05/03/2016

Abstract

Cloud computing environment provides high scalable resources to the users. Services are shared by the public cloud environment. Private cloud model provide shared data for the users. Mixed cloud architecture is build to access the private cloud data under the public cloud services. Big data are managed and distributed over the cloud resources. High scalable data analysis requires huge amount of computational resources. Service composition methods are adapted to fetch the suitable service providers from the public cloud for the private cloud data access. History records are analyzed to identify the better service providers with reference to their performance levels. Privacy and security are provided for the big data and history records. Data classification operations are also carried out on the big data values. K means clustering algorithm, data aggregation methods and K anonymity methods are employed in the history records analysis framework. Bayesian classification algorithm is applied for the data classification process. The system also integrates the Map reduce model for the task and data partitioning process. Accuracy level is increased with privacy and security features.

Reviewed by* **ICETSET'16 *organizing committee*

Index Terms: cloud ,service composition ,Big data ,Map Reduce, History Record.

1. Introduction

Cloud computing, a disruptive trend at present, poses a significant impact on current IT industry and research communities. Cloud computing provides massive computation power and storage capacity via utilizing a large number of commodity computers together, enabling users to deploy applications cost-effectively without heavy infrastructure investment. Cloud users can reduce huge upfront investment of IT infrastructure and concentrate on their own core business. Numerous potential customers are still hesitant to take advantage of cloud due to privacy and security concerns. The research on cloud privacy and security has come to the picture. Privacy is one of the most concerned issues in cloud computing and the concern aggravates in the context of cloud

computing although some privacy issues are not new. Personal data like electronic health records and financial transaction records are usually deemed extremely sensitive although these data can offer significant human benefits if they are analyzed and mined by organizations such as disease research centre. For instance, Microsoft Health Vault, an online cloud health service, aggregates data from users and shares. The data with research institutes. Data privacy can be divulged with less effort by malicious cloud users or providers because of the failures of some traditional privacy protection measures on cloud. This can bring considerable economic loss or severe social reputation impairment to data owners. Hence, data privacy issues need to be addressed urgently before data sets are analyzed or shared on cloud.

Data anonymiztion has been extensively studied and widely adopted for data privacy preservation in non interactive data publishing and sharing scenarios. Data anonymization refers to hiding identity and/or sensitive data for owners of data records. Then, the privacy of an individual can be effectively preserved while certain aggregate information is exposed to data users for diverse analysis and mining. A variety of anonymization algorithms with different anonymization operations have been proposed. Large-scale data processing frameworks like MapReduce have been integrated with cloud to provide powerful computation capability for applications. So, it is promising to adopt such frameworks to address the scalability problem of anonymizing large-scale data for privacy preservation. In our research, we leverage MapReduce, a widely adopted parallel data processing framework, to address the scalability problem of the top-down specialization (TDS) approach for large scale data anonymization. The TDS approach, offering a good trade-off between data utility and data consistency, is widely applied for data anonymization. Most TDS algorithms are centralized, resulting in their inadequacy in handling large scale data sets. Although some distributed algorithms have been proposed, they mainly focus on secure anonymization of data sets from multiple parties, rather than the scalability aspect. As the MapReduce computation paradigm is relatively simple, it is still a challenge to design proper MapReduce jobs for TDS.

2. Related Work

MapReduce is popularized by Google as a very simple but powerful program model that offers parallelized computation fault-tolerance and distributing data processing. Its open-source implementation, Hadoop, provides a software framework for distributed processing of large datasets. We review related work in a number of directions. MapReduce Performance Tuning: Several studies were published on tuning the performance of MapReduce. These include tuned different parameters of Hadoop MapReduce for performance. Dai et al. developed Hi Tune for Hadoop performance analysis and tuning. Herodotou et al. designed a cost-based optimizer with performance knobs to help choose better Hadoop configurations. Zaharia et al. proposed a new scheduling algorithm, called Longest Approximate Time to End (LATE), for environments with heterogeneous server configurations. An anthanarayanan et al. proposed Mantri monitor tasks and culls outliers for better job completion time and later proposed Scarlett replicate data blocks to alleviate hotspots. Jahani et al. applied compiler techniques for Hadoop optimizations. Tan et al. documented and extensively analyzed the performance problem of delay tails

in Hadoop Map Reduce programs caused by long Reduce Tasks. None of these works investigated the I/O problem caused by MapReduce intermediate data shuffling. Our work takes on a different perspective to investigate new strategies for efficient data movement in Map Reduce, relieving its I/O contention.

Map Reduce Data Communication: Kim et al. improved the performance of Map Reduce by reducing redundant I/O in the software architecture. But it did not study the I/O issue caused by data shuffling between Map Tasks and Reduce Tasks. The closest work to our project is Map Reduce Online as proposed by Condie et al. This work focused on enabling instant shuffling of intermediate data from MapTasks to Reduce Tasks. MapReduce Online introduces direct data shuffling channels between Map Tasks and Reduce Tasks to avoid the creation of intermediate Map Output Files. In doing so, it requires the direct coupling of each Map Task with all Reduce Tasks and completely changes the fault handling mechanism of Hadoop. A failure of a Map Task or a Reduce Task is no longer a local event that can easily be recovered by re-launching the failed task. In addition, MapReduce Online requires a large number of sustained TCP connections, which severely limits its scalability. In contrast, our work does not require close coupling of data flow between MapTasks and ReduceTasks, thus allowing separated recovery from failures of either MapTasks or ReduceTasks.

Spark is an emerging MapReduce-based system for big data analytics. It recognizes the disk I/O bottleneck issue during the data shuffling and relaxes the sorting/merging requirement at the reduce sides. i.e., it is not necessary to sort intermediate data before a ReduceTask starts processing them. By relaxing such constraint, data shuffling and computation can be pipelined and accomplished in memory. Spark requires very high memory consumption for shuffling and merging in memory. In addition, by retaining intermediate results in memory, Spark can efficiently accelerate many data-intensive programs, such as weather prediction applications that require iterative algorithms.

Power and Energy of MapReduce Programs: Leverich et al. modified Hadoop to allow scale-down of operational clusters which could save between 9% and 50% of energy consumption. They also outlined further research into the energy-efficiency of Hadoop. Lang et al. closely examined two techniques, namely Covering Set (CS) and All-In Strategy (AIS), which could be used for the management of MapReduce clusters. They showed that AIS was the right strategy for energy conservations. Chen et al. presented a statistics-driven workload generation framework which distilled summary statistics from production MapReduce traces and realistically reproduced representative workloads. This methodology could be useful for understanding design trade-offs in MapReduce. The same team also exploited and analyzed how compression could improve performance and energy efficiency for MapReduce workloads. They proposed an algorithm that examines per job data characteristics and I/O patterns and decides when and where to use compression. Our work does not directly study energy conservation techniques, but evaluates the benefits of virtual shuffling in energy savings. This is complementary to previous research efforts. Our work documents a case study in conserving energy by reducing other related system activities such as disk access

3. Service Selections Schemes

In recent years, Cloud Computing and big data receives enormous attention internationally due to various business-driven promises and expectations such as lower upfront IT costs, a faster time market and opportunities for creating value-add business. As the latest computing paradigm, cloud is characterized by delivering hardware and software resources as virtualized services by which users are free from the burden of acquiring the low level system administration details. Cloud computing promises a scalable infrastructure for processing big data applications such the analysis of huge amount of medical data. By leveraging Cloud services to host Web, big data applications can benefit from cloud advantages such as elasticity, pay-per-use and abundance of resources with practically no capital investment and modest operating cost proportional to actual use.

In practice, to satisfy different security and privacy requirements, cloud environments usually consist of public clouds, private clouds and hybrid clouds, which lead a rich ecosystem in big data applications. Generally, current implementations of public clouds mainly focus on providing easily scaled up and scaled-down computing power and storage. If data centers or domain specific services center tend to avoid or delay migrations of themselves to the public cloud due to multiple hurdles, from risks and costs to security issues and service level expectations, they often provide their services in the form of private cloud or local service host. For a complex web-based application, it probably covers some public clouds, private clouds or some local service host. For instance, the healthcare cloud service, a big data application illustrated, involves many participants like governments, hospitals, pharmaceutical research centers and end users. As a result, a healthcare application often covers a series of services respectively derived from public cloud, private cloud and local host.

In practice, some big data centers or software services cannot be migrated into a public cloud due to some security and privacy issues. If a web based application covers some public cloud services, private cloud services and local web services in a hybrid way, cross-cloud collaboration is an ambition for promoting complex web based applications in the form of dynamic alliance for value-add applications. It needs a unique distributed computing model in a network-aware business context.

Cloud computing environment provides scalable infrastructure for big data applications. Cross clouds are formed with the private cloud data resources and public cloud service components. Cross cloud service composition provides a concrete approach capable for large scale big data processing. Private clouds refuse to disclose all details of their service transaction records. History record based Service optimization method (HireSome-II) is privacy aware cross cloud service composition method. QoS history records are used to estimate the cross cloud service composition plan. K-means algorithm is used as a data filtering tool to select representative history records. HireSome-II reduces the time complexity of cross cloud service composition plan for big data processing. The following drawbacks are identified from the existing system. The following issues are identified from the current cross cloud service composition methods. Big data processing is not integrated with the system. Security and privacy for big data is not provided. Limited scalability in big data process. Mining operations are not integrated with the system

4. Big data Analysis Using Map Reduce

Today huge amount of digital data is being accumulated in many important areas, including e-commerce, social network, finance, health care, education and environment. It has become increasingly popular to mine such big data in order to gain insights to help business decisions or to provide better personalized, higher quality services. In recent years, a large number of computing frameworks have been developed for big data analysis. Among these frameworks, MapReduce is the most widely used in production because of its simplicity, generality and maturity. We focus on improving MapReduce in this paper.

Big data is constantly evolving. As new data and updates are being collected, the input data of a big data mining algorithm will gradually change and the computed results will become stale and obsolete over time. In many situations, it is desirable to periodically refresh the mining computation in order to keep the mining results up-to-date. For example, the PageRank algorithm computes ranking scores of web pages based on the web graph structure for supporting web search. The web graph structure is constantly evolving; Web pages and hyper-links are created, deleted and updated. As the underlying web graph evolves, the PageRank ranking results gradually become stale, potentially lowering the quality of web search. Therefore, it is desirable to refresh the PageRank computation regularly.

Incremental processing is a promising approach to refreshing mining results. Given the size of the input big data, it is often very expensive to rerun the entire computation from scratch. Incremental processing exploits the fact that the input data of two subsequent computations A and B are similar. Only a very small fraction of the input data has changed. The idea is to save states in computation A, re-use A's states in computation B and perform re-computation only for states that are affected by the changed input data. A MapReduce program is composed of a Map function and a Reduce function. Their APIs are as follows:

$$\textbf{Map(K1, V1)} \rightarrow \textbf{[<K2, V2>]}$$

$$\textbf{Reduce(K2, \{V2\})} \rightarrow \textbf{[<K3, V3>]}$$

The Map function takes a kv-pair <K1, V1> as input and computes zero or more intermediate kv-pairs <K2, V2>s. Then all <K2, V2> is grouped by K2. The Reduce function takes a K2 and a list of {V2} as input and computes the final output kv-pairs <K3, V3> s.

A Map Reduce system usually reads the input data of the Map Reduce computation from and writes the final results to a distributed file system, which divides a file into equal-sized blocks and stores the blocks across a cluster of machines. For a Map Reduce program, the Map Reduce system runs a Job Tracker process on a master node to monitor the job progress and a set of Task Tracker processes on worker nodes to perform the actual Map and Reduce tasks. The Job Tracker starts a Map task per data block and typically assigns it to the Task Tracker on the machine that holds the corresponding data block in order to minimize communication overhead. Each Map task calls the Map function for every input <K1, V1> and stores the intermediate kv-pairs <K1, V1> s on local disks. Intermediate results are shuffled to reduce tasks according to a partition function on K2. After a Reduce task

obtains and merges intermediate results from all Map Tasks, it invokes the Reduce function on each <K2, V2> to generate the final output KV-pairs <K3, V3> s.

5. Privacy Ensured High Scalable Data Analysis

History record based Service optimization method (HireSome-II) is enhanced to process big data values. Security and privacy is provided for cross cloud service composition based big data processing environment. Privacy preserved map reduce methods are adapted to support high scalability. The HireSome-II scheme is upgraded to support mining operations on big data.

5.1 Data Analysis

Security and privacy preserved big data processing is performed under the cross cloud environment. Big data classification is carried out with the support of map reduce mechanism. Service composition methods are used to assign resources. The system is divided into six major modules. They are Cross Cloud Construction, Big Data Management, History Analysis, Map Reduce Process, Service Composition and Big Data Classification. Public and private clouds integrated in the cross cloud construction process. Big data management module is designed to provide big data for the cloud users. Resource sharing logs are analyzed under the history analysis. Task partitions operations are performed under the map reduce process. Service provider selection is carried out service composition module. Classification process is carried out under the cross cloud environment.

Private and public cloud resources are used in the cross cloud construction process. Big data values are provided under the data centers in private cloud environment. Service components are provided from public cloud environment. Public cloud services utilize the private cloud data values. Larger and complex data collections are referred as big data. Medical data values are represented in big data form. Anonymization techniques are used to protect sensitive attributes. Big data values are distributed with reference to the user request. Service provider manages the access details in the history files. User name, data name, quantity and requested time details are maintained under the data center. History data values are released with privacy protection. Data aggregation is applied on the history data values. Map reduce techniques are applied to break the tasks. Map reduce operations are partitioned with security and privacy features. Redundancy and fault tolerance are controlled in the system. The data values are also summarized in the map reduce process.

HireSome-II scheme is adapted for the service opus process. History records are analyzed with K-means clustering algorithm. Privacy preserved data message is employed in the system. Public cloud service components are provided to the big data process. Medical data analysis is carried out on the cross cloud environment. Privacy preserved data sorting is applied on the medical data values. Public cloud resources are allocated for the sorting process. Bayesian algorithm is tuned to perform data sorting on parallel and distributed environment.

6. Future Enhancement

In this adoption of our approach to the bottom-up generalization algorithms for data anonymization.

Optimized balanced scheduling in this paper, an enhanced History record-based Service optimization method, for future work we plan to apply our method to scalable privacy preservation aware data set scheduling. Through implementation and simulation, we show some specific cloud systems for processing big data applications. We will investigate the strategies are expected to be developed towards overall that our solution is both scalable and efficient.

7. Conclusion

Service composition methods are used to provide resources for big data process. History record based Service optimization method (HireSome-II) is used as privacy ensured service composition method. HireSome-II scheme is enhanced with privacy preserved big data process mechanism. Map reduce techniques are also integrated with the HireSome-II scheme to support high scalability. Security and privacy are provided for the big data and history data values under the cloud environment. Map reduce techniques reduces the computational complexity in big data processing. Data classification is performed on sensitive big data values with cloud resources. Efficient resource sharing is performed under cross cloud environment.

References

[1] S. Chaudhuri, "What Next?: A Half-Dozen Data Management Research Goals for Big Data and the Cloud," Proc 31st Symp. Principles of Database Systems (PODS '12), pp. 1-4, 2012.

[2] M. Zaharia, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker and I. Stoica,"Resilient distributed datasets: A fault-tolerant abstraction for, in-memory cluster computing," in Proc. 9th USENIX Conf. Netw. Syst. Des. Implementation, 2012, p. 2.

[3] L. Wang, J. Zhan, W. Shi and Y. Liang, "In Cloud, Can Scientific Communities Benefit from the Economies of Scale?," IEEE Trans. Parallel and Distributed Systems, Feb. 2012.

[4] N. Mohammed, B.C. Fung and M. Debbabi, "Anonymity Meets Game Theory: Secure Data Integration with Malicious Participants," VLDB J., vol. 20, no. 4, pp. 567-588, 2011.

[5] D. Zissis and D. Lekkas, "Addressing Cloud Computing Security Issues," Future Generation Computer Systems, vol. 28, no. 3, 2011.

[6] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola and J. M. Hellerstein, "Distributed graphlab: A framework for machine learning and data mining in the cloud," in Proc. VLDB Endowment, 2012.

[7] L. Hsiao-Ying and W.G. Tzeng, "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding," IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 6, 2012.

[8] Xuyun Zhang, Laurence T. Yang, Chang Liu and Jinjun Chen, "A Scalable Two-Phase Top-Down Specialization Approach for Data Anonymization Using MapReduce on Cloud", IEEE Transactions On Parallel And Distributed Systems, Vol. 25, No. 2, February 2014

[9] Yanfeng Zhang, Shimin Chen, Qiang Wang and Ge Yu, "i2MapReduce: Incremental MapReduce for Mining Evolving Big Data", IEEE Transactions On Knowledge And Data Engineering, Vol. 27, No. 7, July 2015 Microsoft HealthVault, http://www.microsoft.com/health/ww/products/Pages /healthvault.aspx, 2013.

[10] Weikuan Yu, Yandong Wang, Xinyu Que and Cong Xu, "Virtual Shuffling for Efficient Data Movement in MapReduce", IEEE Transactions On Computers, Vol. 64, No. 2, February 2015.