International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-3, Issue-3, May-2015

Ensuring Security and Completeness of Query Results in Cloud

Sharayu Anap

Abstract— **Cloud technology provides data storage services to users to store large amount of data and also provides services that perform data processing. While data comes from different servers on cloud and the processing of the data is again done on different server, users must be assured about the integrity of the query results. Various applications need to perform the join operation on the data fetched from different data storage servers and send the join result to the client. The integrity methods described provide a way for the user to verify whether query results obtained after retrieving data are correct that no valid data is missing. Integrity check ensures that the data is complete and authentic. Encryption provides data protection for the query results.**

Index Terms— **Integrity, cloud, data storage, security, query result, join operation.**

I. INTRODUCTION

The Cloud technology has made a mark by providing high computational and data storage services. Cloud technology offers services hosted over the internet. It is advantageous as it has several benefits like "pay per use", on demand services, elasticity (scale up or scale down computing requirements). There are three types of clouds viz. private, public and hybrid. The services provided by Cloud can be divided as Iaas (infrastructure as a service), Paas (platform as a service), Saas (software as a service), etc. [12]. The Cloud technology offers separate services for data storage, while the queries on the data can be processed separately. While a separate server is dedicated to process the queries on the data coming from separate storage servers, user must be assured that the query results are correct and no valid tuple is left out. Integrity check provides the user to verify whether the query result is complete or not. The algorithm to check integrity includes various techniques that can be used to check whether the query results are complete and no valid tuple is missing. Along with this, the algorithm also includes encryption and decryption which offers security to the data.

II. LITERATURE SURVEY

The data security and data privacy problems have been addressed in earlier solutions. Considering database as a service various challenges need to be addressed like remote access to data, user interface for database as a service and data privacy. Various encryption techniques like software level encryption and field level encryption are described for addressing data privacy in such an internet-based database as a service. Query rewriting is also described to improve software level encryption. Also, hardware level encryption and row level encryption are described [1].

Y. Yang et al. [2] describe work on authenticated join processing in databases that are outsourced. They have described various algorithms for authenticated join processing based on the availability of the authenticated data structure. They have described three algorithms Authenticated Indexed Sort Merge Join (AISM), Authenticated Indexed Merge Join (AIM), Authenticated Sort Merge Join (ASM) based on the sort and merge. AISM is based on use of single authenticated data structure on the join attribute; AIM needs authenticated data structure on both relations whereas, ASM does not depend on authenticated data structure. M. Xie et al. [3] describe the problem of integrity auditing of the databases which are outsourced to third party service providers. They describe deterministic and probabilistic approaches. Using these approaches, small numbers of records are inserted into the outsourced database for auditing the integrity of the query results. Using the deterministic function, the client can keep track of the fake tuples in the outsourced data for checking the query integrity.

Earlier solutions deal with the problem of providing data confidentiality and data integrity. Earlier work describes indexing techniques which is used for accessing outsourced data. Encryption of the databases provides data security. The queries are executed on the encrypted data. For making this possible, indexing information is also stored [4]. The problem of integrity for join queries is addressed by using various techniques like markers, twins, salts and buckets in case of join queries [5].

S. De Capitani di Vimercati et al. [6] address the problem of providing access control for outsourced data to honest but curious servers. The solution combines authorization and encryption. They describe a two layer encryption approach for data outsourcing and management of authorization policy. E. Mykletun et al. [7] investigate on the problem of ensuring data integrity in case of outsourced database model where organizations outsource data to external service providers. They suggest various schemes for authentication of query results.

D. Kossmann et al. [8] study different architectures for transaction processing in the cloud environment. They list architectures for database applications in cloud and evaluation of services that use the described architectures. The alternative services vary in cost and performance.

Manuscript received May 23, 2015.
Sharayu Anap, Department of Computer Engineering, MET's IOE, BKC; Savitribai Phule Pune University, Pune, Maharashtra, India

Schemes are introduced where users can verify whether their query answers are complete and authentic [9]. C. Curino et al. [10] introduce Relational Cloud which is a relational database-as-a-service for environments like cloud computing. Relational cloud uses encryption for database privacy.

Map/Reduce is used for processing large amounts of data to produce various kinds of derived data. Join algorithms describe use of Map/Reduce for evaluating join operations. Join algorithms include two-way join and multi-way join categories [11].

III. Security and Integrity Check Methodology

The use of cloud technology enables the integration of various services that store data and that perform query execution. Various methods can be used for providing integrity on the join query results. The user sends a query having join operation and the user may not have any information about where the data is stored and which servers are performing execution of the join queries. The user's request of the join query is sent to the query execution engine which in turn sends sub-queries to the servers storing data. The results of the sub-queries are then returned to the execution server which then performs the join operation on the two sub-query results and returns the final result to the user.

While performing the join various approaches for providing integrity check are described like Markers, Twins, Salts and Buckets. Encryption is used for data protection. The integrity check system enables the user to verify the completeness of the results obtained after executing the join query on data obtained from different storage servers [5]. The different modules of the Integrity check system are shown in Fig 1. The client system sends the query to perform join operation to the query processing server. The query processing server sends the subquery to the data storage servers. The data storage servers execute the subquery. Then additional fake tuples or duplicate tuples are inserted into the result based on the integrity check algorithm. After inserting the additional tuples, encryption is performed on the result for each tuple. The encrypted data is then sent to the query processing server. The query processing server performs the join on both the subquery results as input. It returns the results which are also in encrypted form to the client. At the client, the results are decrypted first. Then integrity check is performed to check for the additional tuples. If integrity check is successful i.e. the results are correct then the additional fake tuples are removed and the actual results are obtained. If there is integrity error or the results appear to be incomplete then integrity error is displayed.

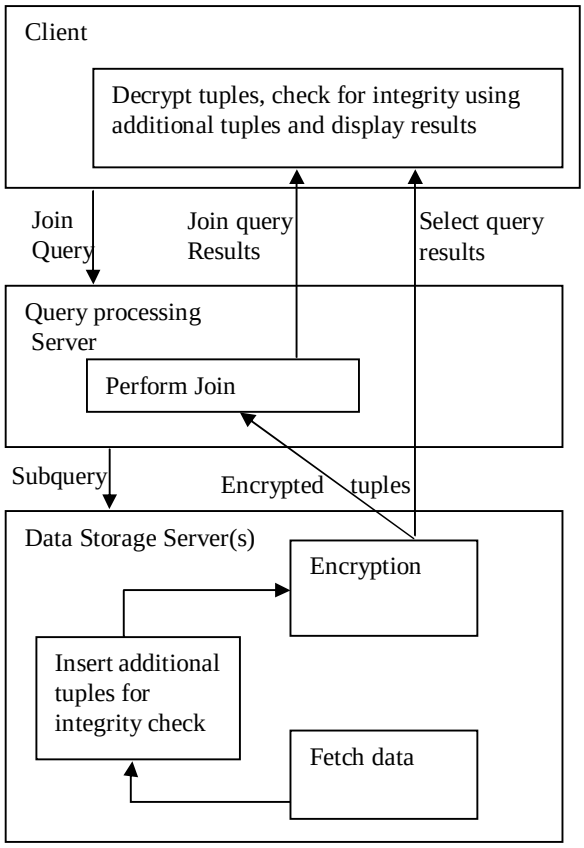

Fig 1: Flow for query evaluation after adding integrity and security mechanisms

Algorithm

The algorithm for verifying the integrity of join query result provides methods for checking the completeness of join results and provide data protection on results [5]. By adding mapping, sorting and reducing steps, the performance of the join processing can be improved. The algorithm proceeds as follows:

1. Client sends the encrypted query and subqueries to the query processing server with the encrypted details of the additional tuples to be added for checking completeness of results.
2. Get the encrypted query from client at the Query Processing Server and send it to the data storage servers.
3. Decrypt the subqueries, details of additional tuples for integrity check and fetch the subquery results from the Data Storage Server(s).
4. Insert additional tuples as per the details provided by client.
5. Apply encryption to all the tuples of the subquery results and send to the Query Processing Server.
6. At the Query Processing Server; return the select query results to client, if the requested query is a select query. If requested query is the join query then perform the join operation on the subquery results.

7. In case of join operation, it can be performed using mapping, sorting and reducing method to improve performance using following steps:
   - Extract unique tuples
   - Sort the extracted tuples.
   - Get all the tuples from second subquery result that match the tuple in the first subquery result. Repeat this for all tuples in first subquery result.
8. Return the results of join query operation to the client system. In case of select query, return the results of the select query to the client system.
9. After receiving query results from Query Processing Server, decrypt the results, check for the integrity by verifying the additional tuples at the client.
10. Get the actual results by identifying the additional tuples and removing them at the client.
11. If integrity check is successful then display the query results at the client. Display message in case of integrity error after verifying additional tuples if the complete query results are not obtained.

IV. Experimental Setup and Result Analysis

The Integrity check is performed involving various modules like fetching of data, inserting additional tuples, encryption; join query processing, decryption and integrity check and display of join result. Fig 2 shows a graph of the time required (in milliseconds) for performing regular join operation and time required for join by using mapping, sorting and reducing concept. Both the join operations are performed after carrying out various steps like fetching of data, inserting additional tuples for integrity check and encrypting all the tuples. The results are obtained by performing one-to-one join operation on around 400 tuples.

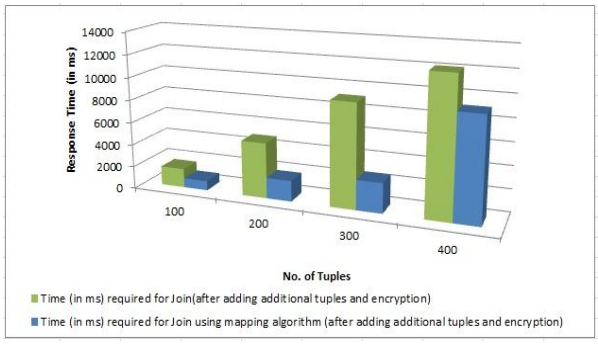

Fig 2: Response time (in milliseconds) with no. of tuples

Table 1 shows the details of the graph (Fig 2) based on number of tuples. The time required to perform join using the mapping, sorting and reducing method is comparatively less than time required for performing normal join as it identifies and extracts only unique tuples. Thus, the integrity method of duplicating tuples cannot be used while performing join using mapping, sorting, reducing method as it only extracts unique tuples. However random tuples can be added for integrity check.

Table 1: Response time (in milliseconds) with number of tuples

No. of Tuples	Time required for Join(after adding additional tuples and encryption)	Time required for Join using mapping, sorting, reducing algorithm
100	1686	839
200	4892	1830
300	9159	2642
400	12143	9134

Table 2 shows the input, steps and expected results for the regular join operation and join operation after applying map-reduce concept.

Table 2: Comparison of approaches used for Join operation

Sr. No.	Input	Steps	Expected Result
1	Query	-add fake tuples - encrypt data - perform regular join	-result of query -integrity error returned if any tuple is omitted
2	Query	-add fake tuples - encrypt data - perform join using mapping, sorting, reducing algorithm	-result of query is returned faster as compared to normal join operation as only distinct tuples are taken -integrity error returned if any tuple is omitted

Table 3: Integrity Comparison

Type	Error Response
Base Join	No error responses in case of tuple deletion or omission as no integrity check mechanisms are applied. No data security as query result is not encrypted.
Join with additional tuples for integrity check	Integrity error displayed in case of omission of tuples. More the number of additional tuples inserted, more is the accuracy of detecting the omission of tuples. Also data is secure due to encryption.

Table 3 shows the accuracy factor by comparing the regular join operation and integrity mechanisms. It shows that there is no error response while performing regular join. However after applying integrity mechanisms we can get the integrity error response in case of missing tuples.

V. Conclusion

Integrity check methods provide user a way to verify whether the query results are appropriate and correct. They enable the user to check if there is any omission of tuples and also provide data protection by applying encryption and decryption. The mapping, sorting and reducing algorithm performs the join operation in a faster way compared to regular join by considering unique tuples. By providing the

mechanisms to check for integrity, the user is enabled to check for the completeness of the results and encryption makes the data secure and not directly visible. The client can ensure that the query answers obtained are correct and complete for the select queries and one-to-one join operations.

REFERENCES

[1] H. Hacigu ¨mu ¨s, B. Iyer, and S. Mehrotra, "Providing Database As a Service," Proc. 18th Int'l Conf. Data Engineering (ICDE '02), Feb. 2002

[2] Y. Yang, D. Papadias, S. Papadopoulos, and P. Kalnis, "Authenticated Join Processing in Outsourced Databases," Proc. ACM Int'l Conf. Management of Data (SIGMOD '09), June/July 2009.

[3] M. Xie, H. Wang, J. Yin, and X. Meng, "Integrity Auditing of Outsourced Data," Proc. 33rd Int'l Conf. Very Large Data Bases (VLDB '07), Sept. 2007.

[4] A. Ceselli, E. Damiani, S.De Capitani di Vimeecati, S. Jajodia, S. Paraboschi, ad P. Samarati, "Modelling and Assessing Inference Exposure in Encrypted Databases," ACM Trans. Information and System Security, vol. 8, no. 1 , pp. 119-152, Feb. 2005.

[5] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P.Samarati, "Integrity for Join Queries in the Cloud", IEEE Trans. Cloud Computing, vol. 1, no. 2, pp. 187-200, Dec.2013.

[6] S. De Capitani diVimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Encryption Policies for Regulating Access to Outsourced Data," ACM Trans. Database Systems, vol. 35, no. 2, Apr. 2010.

[7] E. Mykletun, M. Narasimha, and G. Tsudik, "Authentication and Integrity in Outsourced Databases," ACM Trans. Storage, vol. 2, no. 2, pp. 107-138, May 2006.

[8] D. Kossmann, T. Kraska, and S. Loesing, "An Evaluation of Alternative Architectures for Transaction Processing in the Cloud," Proc. ACM Int'l Conf. Management of Data (SIGMOD '10), June 2010.

[9] H. Pang, A. Jain, K. Ramamritham and K. Tan, "Verifying Completeness of Relational Query Results In Data Publishing," Proc. ACM Int'l Conf. Management of Data (SIGMOD '05), June 2005.

[10] C. Curino et al., "Relational Cloud: A Database Service for the Cloud," Proc. Fifth Biennial Conf. Innovative Data Systems Research (CIDR '11), Jan. 2011.

[11] Jairam Chandar, "Join Algorithms using Map/Reduce", Masters thesis submitted, school of Computer Science School of Informatics,University of Edinburgh, 2010

[12] http://searchcloudcomputing.techtarget.com/definition/cloud-computing