International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 6 3832 - 3835

Query Result Integrity in Multiple Storage Servers in Distributed Environment

[1]Sharayu Anap [2]Dr. K. V. Metre
Department of Computer Engineering
MET's IOE, BKC
Nasik, India
[1]*sharayuanap@gmail.com*, [2]*kvmetre@gmail.com*

***Abstract*—**The advancements in the internet and cloud technology have led to increasing use of data outsourcing and use of applications provided by different service providers. These services concentrate on their core functionality. Thus they make it possible for the users to use services without adding substantial overhead for hardware and software requirements. If the user wants to execute various queries which would execute on data stored at different servers, then this can be made possible by integration of data obtained from different servers. When the user executes queries on such systems then the main concern remains data security. Using the mechanisms described in the paper, users can be free from concerns of data protection and verification of data obtained from query responses. Providing different ways for checking whether query responses are secure and proper helps users to perform their operations without worrying of the trustworthiness of the servers in public cloud.

Keywords*-query responses; secure data; distributed environment; cloud;*

*****

I. INTRODUCTION

Continued use of cloud technology and distributed systems has made it possible to access services that store data and also perform various operations. Growth in the information technology and networking has encouraged the trend of data outsourcing and data management. Such trends and cloud technology has given rise to various services that can fulfill the data management and storage needs of the customers. Not only database as a service is being used but different servers can be used to perform specific operations like query execution. The separation of services enables to concentrate on core functionality. Thus having separate service for storing data makes data highly available and accessible. In the distributed environment, the data can be distributed and stored at different servers. While performing any query execution, the data may be obtained from different servers and then combined together to obtain the final results. Similarly, having separate service for performing operations make it possible to perform operations at a faster pace. Services offered by cloud could be in terms of platform, infrastructure, software, etc. In case of public cloud the concern of secure data storage is considered important. In case of hybrid cloud, certain services which need to be highly secure might be part of the private cloud and certain services could be part of pubic cloud. There are various types of services offered by the cloud technology viz. platform-as-a-service, infrastructure-as-a-service, and software-as-a-service. Cloud technology offers various advantages in terms of offering various services to the users. The users can gain benefits by paying only as per their usage of the service. Also various resources can be added or minimized as per the demand thus providing elasticity. Cloud allows users to store their data and application on the cloud thereby eliminating the need of individual hardware and software resources [8].

II. LITERATURE SURVEY

There has been a lot of work done related to the security and integrity concerns. Various encryption algorithms have been used to provide secure data at the storage servers. Also various methods have been described for correct results of query evaluation.

S. De Capitani di Vimercati et al. [2] describe the problem of access control for data that is outsourced data. This data is stored on the honest but curious servers. They describe authorization and encryption as a resolution of this problem. A two layer encryption mechanism for data outsourcing and management of authorization policy are described. E. Mykletun et al. [3] highlight on the problem of ensuring data integrity in case of outsourced database model. This is a scenario where organizations outsource data to external service providers. They suggest various schemes that can be used to authenticate the responses of queries. They describe condensed-RSA scheme. Schemes for data integrity for the query results for outsourced databases are mentioned. Their approach is based on signature aggregation mechanisms. In the outsourced data model, the data storage and management are outsourced to the third-party service providers. The integrity needs to be maintained while performing various operations on data such as create, update and access data.

D. Kossmann et al. [4] describe the study of architectures for transaction processing in the cloud environment. The database applications in cloud and services evaluation that use the described architectures are described. The alternative services vary in cost and performance. Schemes where users can verify whether their query answers are complete and authentic are described [5]. C. Curino et al. [6] introduce Relational Cloud, a relational database-as-a-service for cloud computing environments. Relational cloud uses encryption decryption mechanisms for database privacy.

Map/Reduce mechanisms are used for processing large amounts of data in hadoop. It is used to produce various kinds of derived data. Join algorithms can be performed using Map/Reduce techniques. Join algorithms include two-way join and multi-way join categories [7]. P. Devanbu et al. [9] describe integrity of data that is published over the internet. Such information is used for making high value decisions. Hence the publisher of such data must provide the integrity and authenticity of data to the clients. They describe solutions that provide evidence of completeness of query evaluation. Their techniques make use of complex data structures and verification objects.

 ISSN: 2321-8169
3832 - 3835

III. INTERGITY CHECK USING VARIOUS MECHANISMS

The data in the distributed system or in the cloud environment is stored in parts in different servers. Consider a scenario where some tables of a database are stored on one storage server and other tables are stored on second data storage server. If the user wants certain information, various operations can be performed on the database. These operations may include create, update, insert, select for particular records, delete particular records, etc. If the retrieval of the records includes data coming from more than one server, then the data could be accessed and then merged into single result set at a separate server which would be responsible only for executing the queries. While the data is transferred from the data storage servers to the operational server, the server performs the required process and returns the resultant records to the client. However during this process there is a chance of the records getting tampered or deleted from the dataset. When the resultant records are returned to the client, the client may not be able to detect the removal or tampering of the records if any [1]. Fig. 1 shows various components in the cloud. The user system connects to the operational server through the internet and the operational server also connects to the storage servers through the internet.

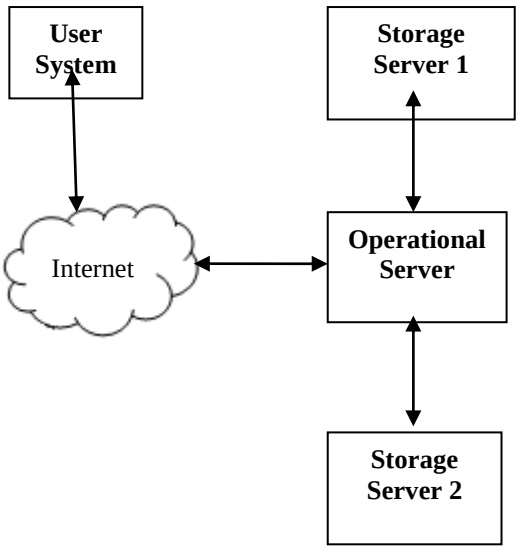

Figure 1. Various components for query processing in cloud.

Hence certain methods can be used so that the client is able to detect such a problem. These techniques include addition of dummy records into the resultant records before sending them to the operational server. The dummy records can be created in various ways. They could be any random values or computed values which do not modify the actual query results. They could be any numbers or strings that can be identified as fake records and can be differentiated from the original records. Such records can be encrypted so that the operational server is not able to differentiate the records and just performs its core task i.e. to execute the query and return the results. One another mechanism for detecting integrity issue is adding of replicated records that are a copy of the original records in the database. All these records can also be encrypted so that the operational server does not have direct visibility of the data. The client provides the information for adding the fake and replicated records to the data storage servers. So that it can verify the same records in the output and can confirm the correctness and completeness of the output values. Using encryption and decryption provides data security and confidentiality [1]. Fig. 2 shows the process of query evaluation in case of multiple data storage servers and a separate operational server.

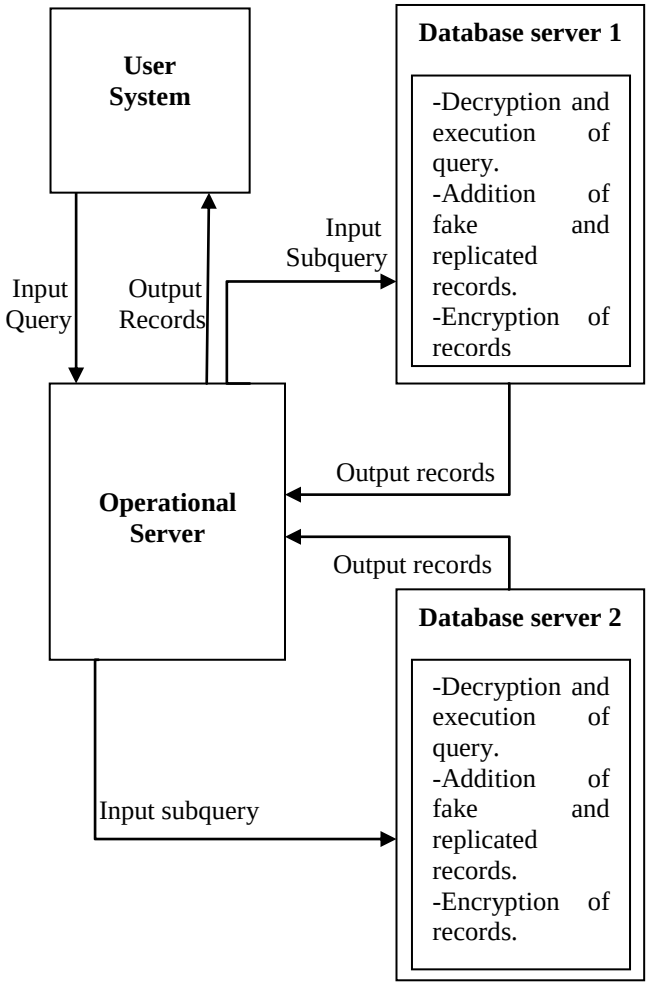

Figure 2. Process for execution of query and output in case of multiple data storages.

A. *Algorithm*

The algorithm describes the steps for query evaluation in case of multiple data storages. It checks for integrity using addition of fake records [1]. The optimization steps are used in case of join query operations. The optimization method describes the steps to perform join operation in an efficient way.

- The client system, operational server and the data storage servers are connected to each other through the internet.
- User sends query to the client system.
- Client system encrypts the query and information about the generation of dummy records and sends the encrypted query to the operational server.
- Operational server sends the input queries and all the related information to the respective database servers.

 ISSN: 2321-8169
3832 - 3835

- Database server decrypts the input query and retrieves all the information for generating fake records.
- It executes the query or sub-query and fetch the resultant records.
- It then generates the dummy records as per the information provided by the client system.
- It then encrypts all the records and sends to the operational server.
- The operational server performs query execution operation and sends the output to the client system.
- Client system decrypts the output records.
- It verifies the dummy values for integrity check and removes those extra records.
- It displays the actual output to the user.
- For performing join operation, extract the unique valued records from the left hand side table of the join query.
- After extracting the records, each tuple is mapped to a key which is the unique join attribute value. So for each table the value matching the join attribute value will be extracted.
- The key value will be matched for the other table in the join query and matching records are extracted.
- Then combine the output records obtained from different tables and return result to client system.

IV. RESULT ANALYSIS

The check for verifying the correctness and completeness of the query output involves various steps like fetching data, addition of fake records or replication of records. Then encryption is performed at record level and then the join operation is performed in case of a join query request by the client. The optimization algorithm perform join in an efficient way as compared to normal join operation. If the client request for simple select query then the resultant records are returned by the operational server to the client. Fig. 3 shows the graph of the time required for performing join operation in a regular way and by using optimized way by varying the number of dummy and replicated records. The time required for the optimized way of performing join is less as compared to that of performing regular join. Table I shows the details of the graph in a tabular format.

The mechanisms that provide a way to assess the query execution results' completeness provide a user with a way to check if any records are missing. Thus even though adding the records for integrity check increases the time for query processing, it shows accuracy of the system in returning the error response for the output to the client. Hence in normal systems there is no error response in case of missing records due to no integrity check mechanisms whereas, the systems having integrity check mechanisms provide user with an error response in case of omission of records.

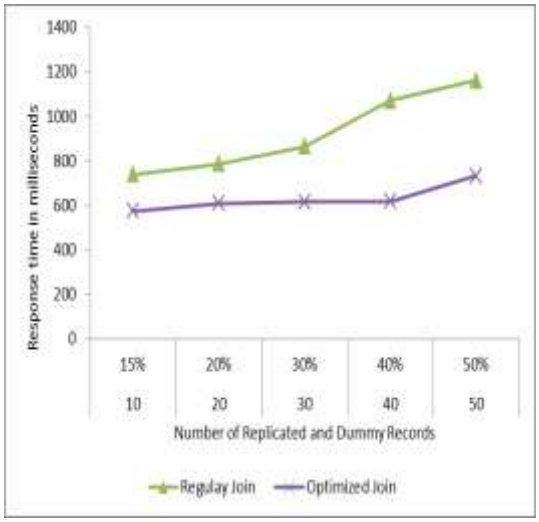

Figure 3. Response time (in ms) of regular join and optimized join with varying no. of fake and replicated records

Table I shows the details of the graph (Fig. 3) in a tabular format. It describes the number of dummy records added and the percentage of replicated records added for 50 original records. It shows the time (in milliseconds) required for performing normal join operation and performing the same operation in a optimized way on these records.

TABLE I. TIME COMPLEXITY IN MILLISECONDS

dummy records	replicated records	regular join	optimized join
10	15%	739	574
20	20%	786	609
30	30%	864	614
40	40%	1070	616
50	50%	1162	731

The graph in Fig. 4 shows the time required for adding varied number of dummy and replicated records and encrypting the complete records. Table II shows the details of the graph (Fig. 4) in a tabular format. It describes the number of dummy records added and the percentage of replicated records added for 50 original records. It shows the time (in milliseconds) required for insertion of dummy and replicated records and performing encryption on all the records.

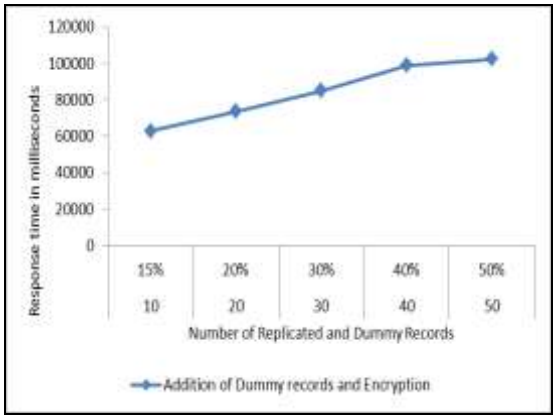

Figure 4. Time (in ms) required for adding dummy and replicated records and encrypting all the records

TABLE II. TIME REQUIRED FOR ADDING DUMMY RECORDS AND ENCRYPTION

Dummy records	Replicated records	Addition of Dummy records and Encryption
10	15%	62731
20	20%	73492
30	30%	84957
40	40%	98778
50	50%	102365

CONCLUSION

The use of multiple data storage servers and use of separate operational server for computationally intensive services has been gaining importance now-a-days. Large storage and access capabilities have made the cloud technology beneficial for fulfilling data storage and computational requirements. Even though use of such scenario is advantageous, it's a concern to provide evidence to the user about the valid and complete results of the query execution answers. The algorithm used for handling such a problem provides ways for verifying the correctness and completeness of the query output. This makes it possible for the user to ensure that the correct and complete output has been received. The dummy records can be easily added and removed adding to some overhead to the actual results. The mechanisms for ensuring integrity can also be further extended for the use of complex queries.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

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

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

[9] P. Devanbu, M. Gertz, C. Martel, and S. Stubblebine, "Authentic Third-Party Data Publication," Proc. IFIP WG11.3 Working Conf. Database and Application Security (DBSec '00), Aug. 2000.