Towards Differential Query Services in Cost-Efficient Clouds

V.Karthik[1], S.Ramesh[2]
[1] *Student, Dept. of MCA, EAIMS*
[2] *Professor, Dept. of MCA, EAIMS, Tirupati, A.P.*

***Abstract*- Cloud computing as an emerging technology trend is expected to reshape the advances in information technology. In a costefficient cloud environment, a user can tolerate a certain degree of delay while retrieving information from the cloud to reduce costs. In this paper, we address two fundamental issues in such an environment: privacy and efficiency. We first review a private keyword-based file retrieval scheme that was originally proposed by Ostrovsky. Their scheme allows a user to retrieve files of interest from an untrusted server without leaking any information. The main drawback is that it will cause a heavy querying overhead incurred on the cloud, and thus goes against the original intention of cost efficiency. In this paper, we present a scheme, termed efficient information retrieval for ranked query (EIRQ), based on an aggregation and distribution layer (ADL), to reduce querying overhead incurred on the cloud. In EIRQ, queries are classified into multiple ranks, where a higher ranked query can retrieve a higher percentage of matched files. A user can retrieve files on demand by choosing queries of different ranks. This feature is useful when there are a large number of matched files, but the user only needs a small subset of them. Under different parameter settings, extensive evaluations have been conducted on both analytical models and on a real cloud environment, in order to examine the effectiveness of our schemes.**

***Index Terms*- purpose of the system, existing system, proposed system, architecture, modules.**

I. INTRODUCTION

Cloud computing is an emerging technology is expected to reshape information technology processes in the near future [1]. Due to the overwhelming merits of cloud computing, e.g., cost-effectiveness, flexibility and scalability, more and more organizations choose to outsource their data for sharing in the cloud. As a typical cloud application, an organization subscribes the cloud services and authorizes its staff to share files in the cloud. Each file is described by a set of keywords, and the staff, as authorized users, can retrieve files of their interests by querying the cloud with certain keywords. In such an environment, how to protect *user privacy* from the cloud, which is a third party outside the security boundary of the organization, becomes a key problem.

User privacy can be classified into *search privacy* and *access privacy* [2]. Search privacy means that the cloud knows nothing about what the user is searching for, and access privacy means that the cloud knows nothing about which files are returned to the user. When the files are stored in the clear forms, a naïve solution to protect user privacy is for the user to request *all* of the files from the cloud; this way, the cloud cannot know which files the user is really interested in. While this does provide the necessary privacy, the communication cost is high.

Private searching was proposed by Ostrovsky et al. [3], [4] (referred to as the Ostrovsky scheme in this paper), which allows a user to retrieve files of interest from an entrusted server without leaking any information. However, the Ostrovsky scheme has a high computational cost, since it requires the cloud to process the query (perform homomorphic encryption) on *every* file in a collection. Otherwise, the cloud will learn that certain files, without processing, are of no interest to the user. It will quickly become a performance bottleneck when the cloud needs to process thousands of queries over a collection of hundreds of thousands of files. We argue that subsequently proposed improvements, like [5], [6], also have the same drawback. Commercial clouds follow a *pay-as-you-go* model, where the customer is billed for different operations such as bandwidth, CPU time, and so on. Solutions that incur excessive computation and communication costs are unacceptable to customers.

To make private searching applicable in a cloud environment, our previous work [7] designed a cooperate private searching protocol (COPS), where a proxy server, called the aggregation and distribution layer (ADL), is introduced between the users and the cloud. The ADL deployed inside an organization has two main functionalities: aggregating user queries and distributing search results. Under the ADL, the computation cost incurred on the cloud can be largely reduced, since the cloud only needs to execute a combined query *once*, no matter how many users are executing queries. Furthermore, the communication cost incurred on the cloud will also be reduced, since files shared by the users need to be returned only once. Most importantly, by using a series of secure functions, COPS can protect user privacy from the ADL, the cloud, and other users.

1.2 Existing System

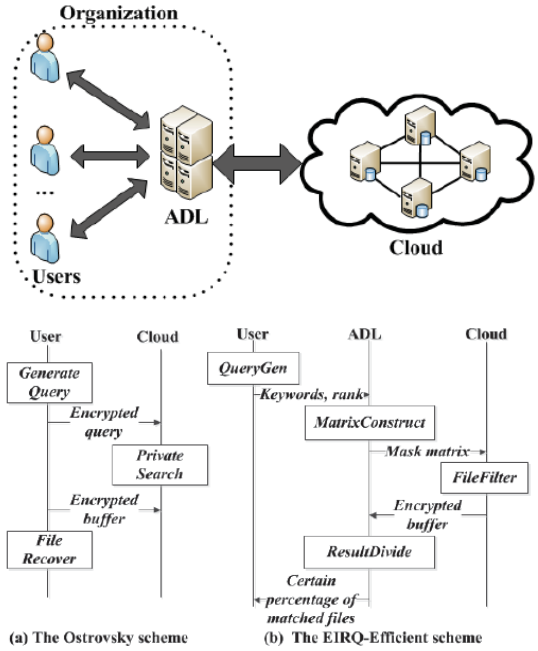
Existing system private keyword-based file retrieval scheme, which was originally proposed by Ostrovsky. Their scheme allows a user to retrieve files of interest from an entrusted server without leaking any information. The main drawback is that it will cause a heavy querying overhead incurred on the cloud, and thus goes against the original intention of cost efficiency.

Private searching was proposed by Ostrovsky et al. which allows a user to retrieve files of interest from an entrusted server without leaking any information. However, the Ostrovsky scheme has a high computational cost, since it requires the cloud to process the query on every file in a collection. Otherwise, the cloud will learn that certain files, without processing, are of no interest to the user. It will quickly become a performance bottleneck when the cloud needs to process thousands of queries over a collection of hundreds of thousands of files.

1.3 Proposed System

We propose a scheme, termed Efficient Information retrieval for Ranked Query (EIRQ), in which each user can choose the rank of his query to determine the percentage of matched files to be returned. The basic idea of EIRQ is to construct a privacy preserving *mask matrix* that allows the cloud to filter out a certain percentage of matched files before returning to the ADL. This is not a trivial work, since the cloud needs to correctly filter out files according to the rank of queries without knowing anything about user privacy. Focusing on different design goals, we provide two extensions: the first extension emphasizes *simplicity* by requiring the least amount of modifications from the Ostrovsky scheme, and the second extension emphasizes *privacy* by leaking the least amount of information to the cloud.

II. ARCHITECTURE DIAGRAM



III. MODULES

The project has been divided into 4 different modules:

1. Differential Query Services
2. Efficient Information Retrieval For Ranked Query
3. Aggregation And Distribution Layer
4. Ranked Queries

3.1 Differential Query Services:

We introduce a novel concept, differential query services, to COPS, where the users are allowed to personally decide how many matched files will be returned. This is motivated by the fact that under certain cases, there are a lot of files matching a user's query, but the user is interested in only a certain percentage of matched files. To illustrate, let us assume that Alice wants to retrieve 2% of the files

that contain keywords "A, B", and Bob wants to retrieve 20% of the files that contain keywords "A, C". The cloud holds 1,000 files, where {F1, . . . , F500} and {F501, . . . , F1000} are described by keywords "A, B" and "A, C", respectively. In the Ostrovsky scheme, the cloud will have to return 2, 000 files. In the COPS scheme, the cloud will have to return 1, 000 files. In our scheme, the cloud only needs to return 200 files. Therefore, by allowing the users to retrieve matched files on demand, the bandwidth consumed in the cloud can be largely reduced.

3.2 Efficient Information Retrieval For Ranked Query:

We propose a scheme, termed Efficient Information retrieval for Ranked Query (EIRQ), in which each user can choose the rank of his query to determine the percentage of matched files to be returned. The basic idea of EIRQ is to construct a privacypreserving *mask matrix* that allows the cloud to filter out a certain percentage of matched files before returning to the ADL. This is not a trivial work, since the cloud needs to correctly filter out files according to the rank of queries without knowing anything about user privacy. Focusing on different design goals, we provide two extensions: the first extension emphasizes *simplicity* by requiring the least amount of modifications from the Ostrovsky scheme, and the second extension emphasizes *privacy* by leaking the least amount of information to the cloud.

3.3 Aggregation And Distribution Layer :

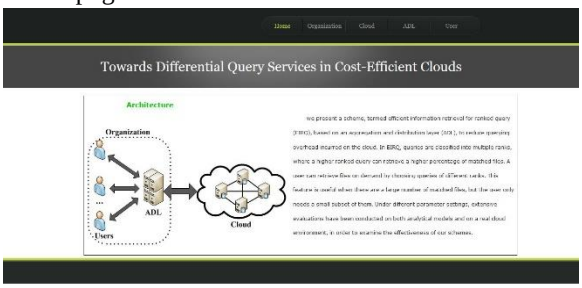
An ADL is deployed in an organization that authorizes its staff to share data in the cloud. The staff members, as the authorized users, send their queries to the ADL, which will aggregate user queries and send a combined query to the cloud. Then, the cloud processes the combined query on the file collection and returns a buffer that contains all of matched files to the ADL, which will distribute the search results to each user. we will discuss the computation and communication costs as well as the querying delay incurred on the ADL.

3.4 Ranked Queries:

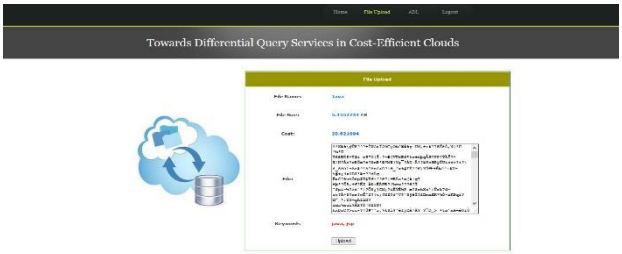
To further reduce the communication cost, a differential query service is provided by allowing each user to retrieve matched files on demand. Specifically, a user selects a particular *rank* for his query to determine the percentage of matched files to be returned. This feature is useful when there are a lot of files that match a user's query, but the user only needs a small subset of them.

IV. SCREENSHOTS

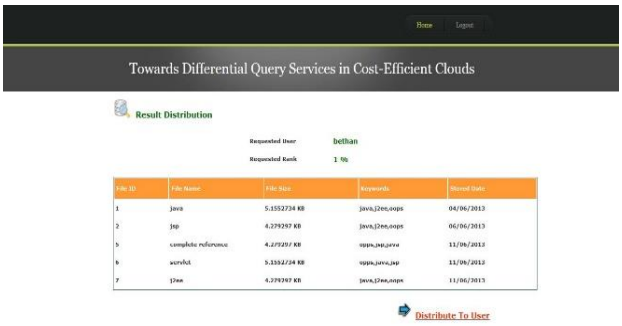
Home page:



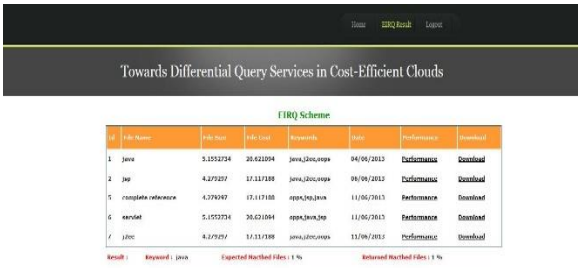
FILE ENCRYPTION:



ADL RETURNING FILES:



USER REQUESTED FILE:



V. CONCLUSION

We proposed three EIRQ schemes based on an ADL to provide differential query services while protecting user privacy. By using our schemes, a user can retrieve different percentages of matched files by specifying queries of different ranks. By further reducing the communication cost incurred on the cloud, the EIRQ schemes make the private searching technique more applicable to a cost-efficient cloud environment. However, in the EIRQ schemes, we simply determine the rank of each file by the highest rank of queries it matches. For our future work, we will try to design a flexible ranking mechanism for the EIRQ schemes.

REFERENCES

References for the project development were taken from the following books and web sites.

[1] P. Mell and T. Grance, "The nist definition of cloud computing (draft)", *NIST Special Publication*, 2011.

[2] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric encryption: improved definitions and efficient constructions", in *Proc. of ACM CCS*, 2006.

[3] R. Ostrovsky and W. Skeith, "Private searching on streaming data," in *Proc. of CRYPTO*, 2005.

[4] "Private searching on streaming data," *Journal of Cryptology*, 2007.

[5] J. Bethencourt, D. Song, and B. Waters, "New constructions and practical applications for private stream searching," in *Proc. Of IEEE S&P*, 2006.

[6] "New techniques for private stream searching," *ACM Transactions on Information and System Security*, 2009.

[7] Q. Liu, C. Tan, J. Wu, and G. Wang, "Cooperative private searching in clouds", *Journal of Parallel and Distributed Computing*, 2012.

[8] G. Danezis and C. Diaz, "Improving the decoding efficiency of Private search," in *IACR Eprint archive number 024*, 2006.

[9] "Space-efficient private search with applications to rate less Codes," *Financial Cryptography and Data Security*, 2007.

[10] M. Finiasz and K. Ramchandran, "Private stream search at the same communication cost as a regular search: Role of ldpc codes", In *Proc. of IEEE ISIT*, 2012.