IISRC International Conference on Industrial Scientific Research Engineering
International Industrial Scientific Research Corporation
www.worldconferences.org

# Cost-Efficient Clouds through Query Differential Services

**MOHAMMED ISHAQ SHERIF[1], CH. SUBBA RAO[2], SYED ABDUL HAQ[3]**
[1]PG Scholar, Dept of CSE, QCET, Nellore, AP, India.
[2]Associate Professor, Dept of CSE, QCET, Nellore, AP, India.
[3]HOD, Dept of CSE, QCET, Nellore, AP, India.

**Abstract:** Cloud computing as an emerging technology trend is expected to reshape the advances in information technology. In a cost efficient 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.

**Keywords:** Efficient Information Retrieval for Ranked Query (EIRQ), Aggregation and Distribution Layer (ADL), Cloud, Searchable Symmetric Encryption (SSE).

## I. INTRODUCTION

The system mainly consists of three entities the aggregation and distribution layer (ADL), many users, and the cloud, as shown in Fig. 1. For simplicity of clarification, we just utilize a solitary ADL in this paper, yet numerous ADLs can be sent as essential. An ADL is sent in an association that approves its staff to share information in the cloud. The staff individuals, as the approved clients, send their questions to the ADL, which will total client inquiries and send a consolidated inquiry to the cloud. At that point, the cloud forms the joined inquiry on the document accumulation and returns a support that contains all of coordinated records to the ADL, which will convey the list items to every client. To total adequate inquiries, the association may require the ADL to sit tight for a span of time before running our plans, which may bring about a sure questioning postponement. In the supplementary record, we will examine the calculation and correspondence costs and additionally the questioning deferral caused on the ADL. To further decrease the correspondence cost, a differential question administration is given by permitting every client to recover coordinated documents on interest. In particular, a client chooses a specific rank for his question to focus the rate of coordinated records to be returned. This element is valuable when there are a great deal of documents that match a client's inquiry, however the client just needs a little subset of them.

## II. LITERATURE SURVEY

**A.''Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions,''**
**AUTHORS: R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky,**

Searchable symmetric encryption (SSE) allows a party to outsource the storage of its data to another party (a server) in a private manner, while maintaining the ability to selectively search over it. This issue has been the center of dynamic examination as of late. In this paper we demonstrate two answers for SSE that at the same time appreciate the accompanying properties: Both arrangements are more productive than all past consistent round plans. Specifically, the work performed by the server per returned report is steady instead of direct in the span of the information. Both arrangements appreciate more grounded security ensures than past consistent round plans. Truth be told, we call attention to inconspicuous however major issues with past ideas of security for SSE, and demonstrate to outline developments which keep away from these pitfalls. Further, our second arrangement likewise accomplishes what we call versatile SSE security, where questions to the server can be picked adaptively (by the enemy) amid the execution of the hunt; this idea is both imperative practically speaking and has not been beforehand considered. Shockingly, regardless of being more secure and more effective, our SSE plans are strikingly basic. We consider the effortlessness of both arrangements as a critical step towards the sending of SSE technologies.As an extra commitment, we additionally consider multi-client SSE. All former take a shot at SSE concentrated on the setting where just the proprietor of the information is fit for submitting pursuit inquiries. We consider the regular augmentation where a self-assertive

gathering of gatherings other than the proprietor can submit seek inquiries. We formally characterize SSE in the multi-client setting, and present an effective development that accomplishes preferred execution over just utilizing access control systems.

**B. "Private Searching on Streaming Data,''**
**AUTHORS: R. Ostrovsky and W. Skeith**

In this paper we consider the problem of private searching on streaming data, where we can efficiently implement searching for documents that satisfy secret criteria (such as the presence or absence of a hidden combination of hidden keywords) under various cryptographic assumptions. Our results can be viewed in a variety of ways: as a generalization of the notion of private information retrieval (to more general queries and to a streaming environment); as positive results on privacy-preserving data mining; and as a delegation of hidden program computation to other machines.

**C. ''Cooperative Private Searching in Clouds,''**
**AUTHORS: Q. Liu, C. Tan, J. Wu, and G. Wang**

With the increasing popularity of cloud computing, there is increased motivation to outsource data services to the cloud to save money. An imperative issue in such a situation is to shield client protection while questioning information from the cloud. To address this issue, specialists have proposed a few methods. Be that as it may, existing strategies cause overwhelming computational and transfer speed related expenses, which will be unsuitable to clients. In this paper, we propose a helpful private looking (COPS) convention that gives the same security assurances as earlier conventions, yet with much lower overhead. Our convention permits numerous clients to join their inquiries to decrease the questioning expense while securing their protection. Broad assessments have been led on both investigative models and on a genuine cloud environment to inspect the adequacy of our convention. Our reproduction results demonstrate that the proposed convention lessens computational expenses by 80% and transfer speed expense by 37%, notwithstanding when just five clients inquiry information.

**D.''Improving the Decoding Efficiency of Private Search''**
**AUTHORS: G. Danezis and C. Diaz**

We show two ways of recovering all matching documents, in the Ostrovsky et al. Private Search while requiring considerably shorter buffers. Both schemes rely on the fact that documents colliding in a buffer position provide the sum of their plaintexts. Efficient decoding algorithms can make use of this property to recover documents never present alone in a buffer position.

## III. EXISTING SYSTEM

Private searching was proposed by Ostrovsky et al. This allows a user to retrieve files of interest from an untrusted server without leaking any information. Something else, the cloud will discover that sure records, without preparing, are of no enthusiasm to the client. Business mists take after a pay-as-you-go model, where the client is charged for diverse operations, for example, transmission capacity, CPU time, et cetera. Arrangements that acquire over the top reckoning and correspondence expenses are unsuitable to clients. To make private seeking material in a cloud situation, our past work outlined a chip in private looking convention (COPS), where an intermediary server, called the conglomeration and dissemination layer (ADL), is presented between the clients and the cloud. The ADL conveyed inside an association has two primary functionalities: conglomerating client inquiries and disseminating indexed lists. Under the ADL, the calculation expense brought about on the cloud can be generally decreased, subsequent to the cloud just needs to execute a consolidated question once, regardless of what number of clients are executing inquiries. Moreover, the correspondence expense brought about on the cloud will likewise be diminished, since documents shared by the clients should be returned just once. Above all, by utilizing a progression of secure capacities, COPS can shield client protection from the ADL, the cloud, and different clients.

**A. Disadvantages of Existing System**

- Ostrovsky plan has a high computational expense, since it obliges the cloud to handle the inquiry on each document in an accumulation.
- It will rapidly turn into an execution bottleneck when the cloud needs to process a huge number of questions over an accumulation of a huge number of documents. We contend that accordingly proposed changes, as additionally have the same disadvantage.

## IV. PROPOSED SYSTEM

In this paper, 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 persuaded by the way that under specific cases, there are a considerable measure of documents coordinating a client's question, yet the client is occupied with just a sure rate of coordinated records In the Ostrovsky plan, the cloud will need to give back 2,000 records. In the COPS conspire, the cloud will need to give back 1,000 documents. In our plan, the cloud just needs to give back 200 records. In this manner, by permitting the clients to recover coordinated records on interest, the transfer speed devoured in the cloud can be to a great extent decreased. Effective Information recovery for Ranked Query (EIRQ), in which every client can pick the rank of his question to focus the rate of coordinated documents to be returned. The essential thought of EIRQ is to build a security protecting veil lattice that permits the cloud to sift through a sure rate of coordinated records before coming back to the ADL. This is not a paltry work, subsequent to the cloud needs to accurately sift through documents as indicated by the rank of questions without knowing anything about client protection.

## V. MODULES

- Differential Query Services:
- Efficient Information Retrieval For Ranked Query:
- Aggregation And Distribution Layer
- Ranked Queries

### A. 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 plan, the cloud will need to give back 2, 000 records. In the COPS conspire, the cloud will need to give back 1, 000 documents. In our plan, the cloud just needs to give back 200 records. In this way, by permitting the clients to recover coordinated documents on interest, the data transfer capacity expended in the cloud can be to a great extent lessened.

### B. Productive Information Retrieval For Ranked Query

We propose a plan, termed Efficient Information recovery for Ranked Query (EIRQ), in which every client can pick the rank of his question to focus the rate of coordinated documents to be returned. The essential thought of EIRQ is to build a security saving cover lattice that permits the cloud to sift through a sure rate of coordinated documents before coming back to the ADL. This is not a minor work, following the cloud needs to accurately sift through documents as per the rank of questions without knowing anything about client security. Concentrating on distinctive outline objectives, we give two expansions: the first augmentation underscores effortlessness by obliging the minimum measure of alterations from the Ostrovsky plan, and the second augmentation stresses protection by releasing the slightest measure of data to the cloud.

### C. Aggregation And Distribution Layer

An ADL is conveyed in an association that approves its staff to share information in the cloud. The staff individuals, as the approved clients, send their inquiries to the ADL, which will total client questions and send a joined inquiry to the cloud. At that point, the cloud forms the joined question on the document gathering and returns a cradle that contains all of coordinated records to the ADL, which will disseminate the list items to every client. To total adequate inquiries, the association may require the ADL to sit tight for a stretch of time before running our plans, which may bring about a sure questioning postponement. In the supplementary record, we will talk about the reckoning and correspondence costs and also the questioning postponement brought about on the ADL.

### D. Ranked Queries

To further decrease the correspondence cost, a differential inquiry administration is given by permitting every client to recover coordinated documents on interest. In particular, a client chooses a specific rank for his inquiry to focus the rate of coordinated documents to be returned. This element is helpful when there are a considerable measure of documents that match a client's question, however the client just needs a little subset of them.

## VI. IMPLEMENTATION AND RESULTS

We will look at three EIRQ plans from the accompanying viewpoints, record survival rate and processing/ correspondence expense acquired on the cloud. At that point, in light of the re enactment results, we send our project in Amazon Elastic Compute Cloud (EC2) to test the move in and exchange out time acquired on the cloud when executing private quests. Note that the energy performance trade-off is crucial to the success of cloud computing, and existing energy-saving techniques are hard to directly extend to a cloud environment As part of our future extensions, we will evaluate the consumed energy overhead in the cloud to verify the effectiveness of our schemes. We use No Rank to denote unranked queries under the ADL.

### A. File Survival Rate

Since queries are classified into 0 to 4 ranks, queries in Rank-0, Rank-1, Rank-2, Rank-3, and Rank-4 should retrieve 100%, 75%, 50%, 25%, 0% of matched files, respectively. However, in Fig. 3, the real failure rate in EIRQ-Simple and EIRQ-Privacy under the Ostrovsky parameter setting is much lower than i/r, and thus, the real file survival rate is higher than the desired value of $1 - i/r$ (about 25% and 50% of files are redundantly returned to users); Only EIRQ-Efficient, which filters a certain percentage of matched files before mapping them to a buffer, provides differential query services. Under the Bloom filter parameter setting, we first obtain corresponding mapping times. Specifically, for file survival rate 100%, 75%, 50%, 25%, we have the optimal mapping times 7, 2, 1, 0.4, respectively. Based on these values, the buffer size can be calculated with Esq. 4-6 for different schemes. In practice, $\gamma$ and $\beta$ must be integers. Thus, we use $\_\gamma\_$ and $\_\beta\_$ to replace the corresponding values. Using these parameters, the file survival rates for different ranks are shown in Fig. 4, where three EIRQ schemes can provide differential query services, and no bandwidth is wasted in each EIRQ scheme. Therefore, in terms of file survival rate, the Bloom filter parameter setting can achieve better performance than the Ostrovsky parameter setting.

### B. Computational Cost

As described in Section 6-(B), the computational cost is mainly determined by the number of exponentiations performed by the cloud, which is almost the same under the Bloom filter and the Ostrovsky parameter settings. In order to justify the analyses, we will compare the computational cost between No Rank and three EIRQ schemes. The comparisons of computational cost on the cloud where the number of queries in each rank ranges from 1 to 25. Under the Bloom filter parameter setting, the computational cost is approximately 14.807s in No Rank, 59.274s in EIRQ Simple, 101.075s in EIRQ-Privacy, and 14.861s in EIRQ Efficient. Under the Ostrovsky parameter setting, the computational cost approximately ranges from 14.8270s to 14.8788s in No Rank, from 59.1671s to 59.3838s in EIRQ-Simple, from

114.0475s to 176.5107s in EIRQ-Privacy, and from 14.8664s to 14.9269s in EIRQ Efficient. In both settings, EIRQ-Privacy consumes the most computation cost, and EIRQ-Efficient, like No Rank, consumes the least computation cost.

### C. Communication Cost

The communication cost mainly depends on the buffer size generated by the cloud, which is calculated in different ways under different parameter settings. Furthermore, the buffer size depends on the number of files that match the queries, which is different when users have different common interests, i.e., the average number of common keywords among user queries. Therefore, in different parameter settings, we will analyze the buffer size under different common interests. Notice that in both settings, EIRQ-Efficient always has the best performance, the next is EIRQ-Privacy, and the last is EIRQ-Simple. Furthermore, EIRQ-Efficient works better than No Rank when only a few users are conducting searches.

## VII. CONCLUSION & FUTURE ENHANCEMENT

In this paper, we proposed three EIRQ plans taking into account an ADL to give differential inquiry administrations while ensuring client protection. By utilizing our plans, a client can recover diverse rates of coordinated records by determining questions of distinctive positions. By further diminishing the correspondence expense brought about on the cloud, the EIRQ plans make the private looking strategy more appropriate to an expense effective cloud environment. Then again, in the EIRQ plans, we basically focus the rank of every document by the most noteworthy rank of questions it coordinates. For our future work, we will attempt to outline an adaptable positioning component for the EIRQ plans.

## VIII. REFERENCES

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing (Draft)," in NIST Special Publication. Gaithersburg, MD, USA: National Institute of Standards and Technology, 2016.

[2] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," in Proc. ACM CCS, 2006, pp.79-88.

[3] R. Ostrovsky and W. Skeith, "Private Searching on Streaming Data," in Proc. CRYPTO, 2005, pp. 233-240.

[4] R. Ostrovsky and W. Skeith, "Private Searching on Streaming Data," J. Cryptol., vol. 20, no. 4, pp. 397-430, Oct. 2007.

[5] J. Bethencourt, D. Melody, and B. Waters, "New Constructions and Practical Applications for Private Stream Searching," in Proc. IEEE SP, 2006, pp. 1-6.

[6] J. Bethencourt, D. Melody, and B. Waters, "New Techniques for Private Stream Searching," ACM Trans. Inf. Syst. Security, vol. 12, no. 3, p. 16, Jan. 2009.

[7] Q. Liu, C. Tan, J. Wu, and G. Wang, "Cooperative Private Searching in Clouds," J. Parallel Distrib. Comput., vol. 72, no. 8, pp. 1019-1031, Aug. 2012.

[8] G. Danezis and C. Diaz, "Improving the Decoding Efficiency of Private Search," Int'l Assoc. Cryptol. Res., IACR Eprint Archive No. 024, Schloss Dagstuhl, Germany, 2006.

[9] G. Danezis and C. Diaz, "Space-Efficient Private Search with Applications to Rateless Codes," in Proc. Money related Cryptogr. Information Security, 2007, pp. 148-162.

[10] M. Finiasz and K. Ramchandran, "Private Stream Search at the Same Communication Cost as a Regular Search: Role of LDPC Codes," in Proc. IEEE ISIT, 2012, pp. 2556-2560.

[11] X. Yi and E. Bertino, "Private Searching for Single and Conjunctive Keywords on Streaming Data," in Proc. ACM Workshop Privacy Electron. Soc., 2011, pp. 153-158.

[12] B. Hore, E.- C. Chang, M.H. Diallo, and S. Mehrotra, "Indexing Encrypted Documents for Supporting Efficient Keyword Search," in Proc. Secure Data Manage., 2012, pp. 93-110.

[13] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," in Proc. EUROCRYPT, 1999, pp. 223-238.

[14] Q. Liu, C.C. Tan, J. Wu, and G. Wang, "Efficient Information Retrieval for Ranked Queries in Cost-Effective Cloud Environments," in Proc. IEEE INFOCOM, 2012, pp. 2581-2585.

[15] S.Yu,C. Wang,K.Ren, andW. Lou, "Achieving Secure, Scalable, Fine-Grained Data Access Control in Cloud Computing," in Proc. IEEE INFOCOM, 2010, pp. 1-9.

[16] G. Wang, Q. Liu, J. Wu, and M. Guo, "Hierarchical Attribute-Based Encryption and Scalable User Revocation for Sharing Data in Cloud Servers," Comput. Security, vol. 30, no. 5, pp. 320-331, July 2011.

[17] M. Mitzenmacher, "Compressed Bloom Filters," IEEE/ACM Trans. Netw., vol. 10, no. 5, pp. 604-612, Oct. 2002.

[18] D. Guo, J. Wu, H. Chen, and X. Luo, "Theory and Network Applications of Dynamic Bloom Filters," in Proc. IEEE INFOCOM, 2006, pp. 1-12.

[19] A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M.Q. Dang, and K. Pentikousis, "Energy-Efficient Cloud Computing," Comput. J., vol. 53, no. 7, pp. 1045-1051, Sept. 2010.

[20] E. Gelenbe, R. Loaned, and M. Douratsos, "Choosing a Local or Remote Cloud," in Proc. NCCA, 2012, pp. 25-30.