Open access Journal International Journal of Emerging Trends in Science and Technology

AADQ's: Advances in Aggregation & Distribution of Query services

Authors
Pavan Kumar J[1], Shanmukha Swamy C.V[2], Priyardarshini M.M[3]
[1]Master of Technology, Department of Computer Science, Shridevi Institute of Engineering and Technology, Tumkur.
Email: *iampavanbdvt@gmail.com*
[2]Assistant Professor, Department of Compuer science, Shridevi Institute of Engineering and Technology, Tumkur.
Email: *Shanmukha.c.v@gmail.com*
[3]Master of Technology, Department of Computer Science, Shridevi Institute of Engineering and Technology, Tumkur.
Email: *priyadarshiniraaghu@gmail.com*

Abstract
Cloud computing has significant efficiency and cost advantages, here is an appealing solution for the cost efficient clouds especially for the pay as you go model. We propose a scheme called Advanced Aggregation & Distribution of query services (AADQ's). In this paper, we address Privacy and Efficiency are the two fundamental issues. Without heavy querying this scheme allows users to retrieve files of their interest without leaking any information which is subject to privacy. It also aggregates user's queries in a short time delay, based on multi keyword and multi ranking techniques, which retrieves higher percentage of matched files. The AADQ contains a caching mechanism which overcomes the drawback of sending redundant queries to the cloud, which intern saves the cost. As it provides an efficient solution to reduce querying overhead incurred on the cloud. This feature is useful when cloud retrieves large numbers of matched files, but the user needs only a few of them. Extensive evaluations have been conducted to show the advantages of this approach.
Keywords: *Multikeyword, Multiranking, Query Services, Caching Mechanism.*

1. Introduction

The most revolutionized computation in the era of internet is cloud computing, the growing popularity of the cloud computing is rapidly increasing because of its other merits, so commercial clouds are taking the advantages of organizations which relies on cloud. Privacy in the cloud includes search and access privacies, cloud don't know anything about, what the user is searching for and what files need to be returned to the user. The ways of searching in the earlier times were requesting all the files from the cloud, where cloud cannot come to know which files are being accessed by the users. It consumes a lot of time and the communication cost will go high. There are other techniques which gives private searching, such as [2] Ostrovsky scheme which has a high computational cost because of [4] homomorphic encryption, as the cloud needs to process thousands of queries for each and every file in the cloud, it will become a performance bottleneck when multiple users are requested. Organizations or customers are being charged by the commercial clouds for the usage of the bandwidth, CPU time, etc. [5] Customers are looking for the solutions of cost efficient clouds. A cooperate private searching protocol is defined as COPS, a proxy server, [1] which aggregates and distributes user queries (Referred to as ADL) the computation cost incurred on the cloud is greatly reduced. As the files need to

be returned only once. Providing differential query services based on the rankings, which further reduced the communication cost where user can find the required files of his interest by providing ranks which retrieves the small subset of files.

This paper introduces an enhanced solution for the ADL scheme called AADQ's, which contains Multikeyword and Multiranking by further extending with a caching mechanism to Differential query services[1]. Motivated by this goal, we propose a scheme termed as *Advances in Aggregation & Distribution of Query services (AADQ's),* As user decides the files need to be returned based on multikeyword and multiranking, files which are being returned to the user are not survived for a long time. If the multiple user requests the same query it will go against both on computation and communication cost, but this can be overcome by using AADQ's caching mechanism where, most command files will survive in the buffer so that user can easily find the ranked files which further reduce the computation and communication cost in the cloud.

2. Related Work

The existing research in the area of private searching, user conducts searches on encrypted data [2],[4],[6] private searching performs keyword-based searches for unencrypted data. In the work of Private searching filters streaming data without compromising any user privacy[7]. It requires the server to return a buffer of files that matches a user's query. Every file is associated with a survival rate, and denotes the probability of file being successfully recovered by the user. In a Paillier cryptosystem the files that mismatch a query will not survive in the buffer, but matched files will have high survival rates. Reference proposed two new communication-optimal constructions; one uses Reed-Solomon codes and allows for a zero-error, and the other is based on irregular LDPC codes and allows for the lower computation cost of the server. The extended types of queries to support disjunctive normal forms (DNF) of keywords, the main drawback of existing private searching schemes is that both the computation and communication costs grow linearly with the number of users executing queries. Thus, when applying these schemes to a large-scale cloud environment, querying costs will be extensive. The previous work was the first to make private searching techniques applied in a cloud environment. However, requires the cloud to return all of the matched files, which may cause a waste of bandwidth when only a small percentage of files are of interest. To overcome this problem, we introduce the concept of AADQ. The main difference between this work is that we provide two extensions to address different aspects of the problem, and we conduct experiments on a real cloud to verify the effectiveness of the proposed approach.

3. Background

3.1 System Model

The model consists of three components such as Users, AADQ and the Cloud which is shown in Fig.1 For ease of explanation, we use only single AADQ, and multiple AADQ's can be deployed inside the organization. Only the authorized users of the organization can request for the file. Users can select [3] multikeyword and multiranking for retrieving files, AADQ needs a very short time delay to aggregate user quarries and returns the matched files to the users. The queries returned to the buffer will survive until the size of the buffer is reached. Buffer automatically empties its contents when it becomes full. It will be more beneficial to the users if they request same query many times. Most used files are stored in the separate buffer, by using the caching mechanism, let's see how the communication and computation cost are incurred on the cloud.

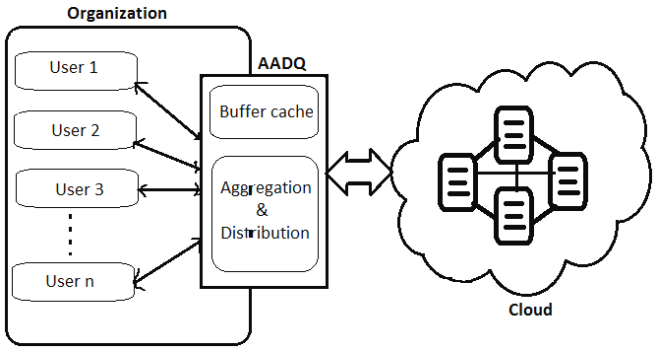

Figure 1 System model.

3.2 Design Goals

All the authorized users of the organization can able to benefit the scheme. AADQ is deployed inside the security boundary of the organization. When all the communication channels are assumed to be secured under existing security protocols during information transfer. Cloud is assumed to be an untrusted, as long as it obeys our scheme. Here cloud itself is the attacker and still it wants to know some more additional information about the user privacy.
The Design goal involves user privacy and cost efficiency.

User privacy: Formally cloud cannot come to know who is requesting for the files and what files to be returned to the user. This involves privacy of searching under multikeyword and multiranking.

Cost efficiency: as it involves a caching mechanism, it retrieves the files from the buffer if it's already been retrieved, otherwise it will aggregate the multiple user queries and sends it to the cloud this retrieves minimal set files on demand.

3.3 Scheme Description

The cloud will construct the buffer size depending upon the size of the files matched, entries of the queries are mapped to the files, and this is done based on different rankings. The mapping is performed by using mask matrix which will further maps the results to the buffer. It takes different states, such as if one file is being matched it will be considered as *survival state*, *more matches* for more than one matched file and *miss a match* for no matches found. More matched files will be able to get top ranks Such as *0*, *1* and the rest will be counted as least matched files.

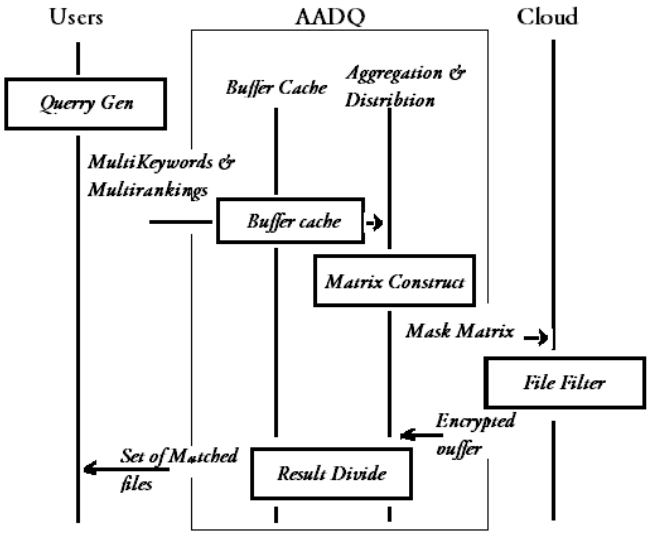

Figure 2: Working process of AADQ.

Based on the homomorphic encryption [4], cloud can able to identify the files which have been recently uploaded by the user, as and when the user establishes a connection by secured communication channels over the cloud, now user can able to communicate cloud via AADQ. When the user seeks files by querying multikeyword and multiranking AADQ will process each user query request with a certain short time delay. If there exists matched files in the buffer cache, it will be retrieved otherwise, aggregated query will be sent to the cloud to retrieve the matched files on demand.
The working process of our scheme is shown in Fig.2 User generates query generation algorithm, if matches found file will be fetched from the buffer cache that runs by buffer cache algorithm, otherwise, AADQ aggregate user queries and sends it to the cloud by running a mask matrix algorithm which constructs masks for user entries of different keywords and rankings. Cloud processes user's requests and filters the matched files by file filter algorithm and sends it to the AADQ. Distributing the matched results will be taken care by using result divide algorithm at the AADQ; it further divides the results for different users.

4. Analysis

We deploy our scheme in Amazon S3 compute cloud (Fig.3) To test transfer in and transfer out time.

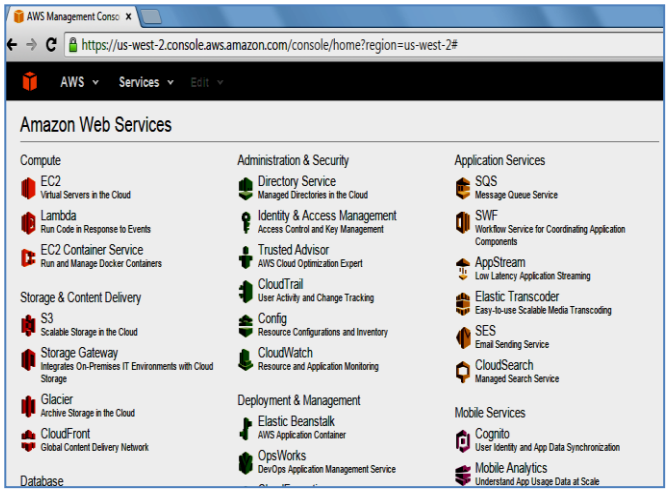

Figure 3: Amazon S3 Real time cloud.

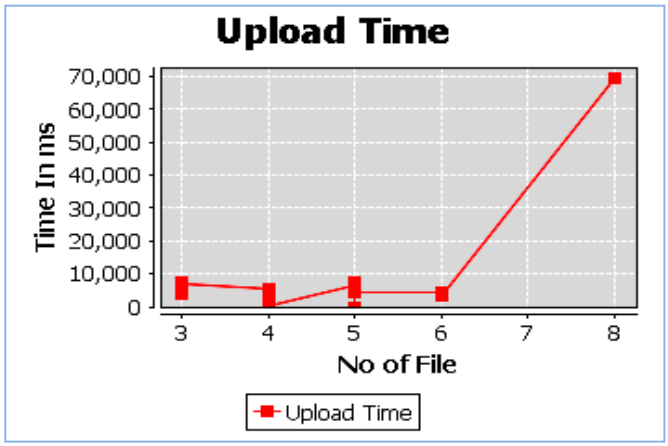

Figure 4: Upload time of file by using AADQ.

Figure 5: Search time of Cloud.

Communication and Computation cost is done in comparison with no rank and different ranking settings we also provide performance of accuracy in comparison with the upload time, search time, Index computation time. This is shown in Fig.4. *X* axis as number of files and *Y* axis as time in milliseconds. Fig.5 shows search time of cloud when files are searched by aggregated queries. Fig.6 the Index computation time which counts the values of multi keywords and multi rankings when it is determined by each user.

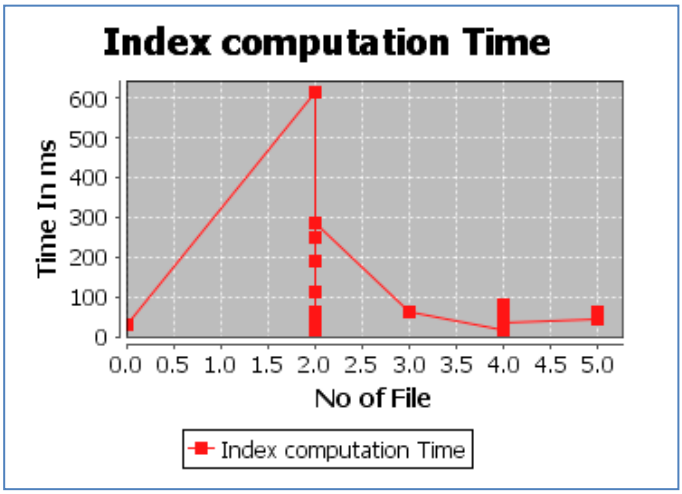

Figure 6: Index Computation time.

The performance of searching and retrieving files when the number of files are uploaded and retrieved is not considerably higher with the implementation of this scheme on real clouds. In future, it can be made more flexible by time efficient.

5. Conclusion

In this paper, we introduce an AADQ technique for searching queries on the cloud, with some specifications of multi keywords and multi rankings technique, which retrieves the minimum set of required files on demand. By providing user privacy which is more cost efficient. This is applicable for the commercial clouds which go against cost computation. In future more flexible and time efficient systems can be designed.

References

1. Qin Liu, Chiu C, Jibe Wu, and Guojun Wang, (2014) Towards Differential Query services in a Cost-Efficient Clouds, ‟IEEE Transactions on parallel and Distributed Systems R.J.SOLOMONOFF,Member,IEEE "Some recent works in Artificial Intelligence," Proceedings of the IEEE,VOL. 54,NO. 12,December 1966.
2. Boone. D, Crescenzo. D, Ostrovsky. R, and Persiano. G, (2004) Public- Key with Keyword Search, ‟ Proc. Int‟l Conf. Theory and Applications of Cryptographic Techniques
3. Cao.N,Wang.C, Ren.M Li, K. And Lou. W, (2011) „Privacy-Preserving Multi keyword Ranked Search over Encrypted Cloud Data, ‟ Proc. IEEEINFOCOM..
4. Coron. J. S, Mandal. A, Naccache. D and Tibouchi. M, (2011) „Fully Homomorphic Encryption over the Integers with ShorterPublic Keys, ‟ CRYPTO‟11: Proc. 31st Ann. Conf. Advances in Cryptology Jong-Myoung Kim, Seon-Ho Park, Young-Ju Han and Tai-Myoung Chung, "CHEF: Cluster Head Election mechanism using Fuzzy logic in Wireless Sensor Networks," ISBN 978-89-5519-136-3,Feb. 17-20, 2008 ICACT 2008.
5. Curtmola. R, Garay. J. A, Kamara. S, and Ostrovsky. R, (2006) „Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions, ‟Proc. ACM 13th Conf. Computer and Comm. Security Junpei Anno,Leonard Barolli, "A Cluster Head
6. Gogle. P, Staddon. J, and Waters. B, (2004) ‟SecureConjunctive Keyword Search over Encrypted Data, ‟ Proc. Second Int‟l Conf. Applied Cryptography and Network Security (ACNS), pp. 31-45.
7. Hu.H, Xu.J, Ren.C, and Choi.B, (2011) „Processing Private Queries over Untrusted Data Cloud through Privacy Homomorphism,‟ Proc. IEEE 27th Int‟l Conf. Data Eng. (ICDE).