International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-3 Number-2 Issue-10 June-2013

A Distributed DB Architecture for Processing cPIR Queries

Sultan.M[1], Karthi.M[2], ManikandaPrabhu.M[3], Muruganandham.N[4], Tashelat Masleena.S[5]

Abstract

Information Retrieval is the Process of obtaining materials, usually documents from unstructured huge volume of data. Several Protocols are available to retrieve bit information available in the distributed databases. A Cloud framework provides a platform for private information retrieval. In this article, we combine the artifacts of the distributed system with Cloud framework for extracting information from unstructured databases. The process involves distributing the database to a number of co-operative peers which will reduce the response of the query by influencing computational resources in the peer. A single query is subdivided into multiple queries and processed in parallel across the distributed sites. Our Simulation results using Cloud Sim shows that this distributed database architecture reduces the cost of computational Private Information Retrieval with reduced response time and processor overload in peer sites.

Keywords

Computational Cloud, Private Information Retrieval, Distributed Systems, Query Systems.

1. Introduction

A Private Information Retrieval (PIR) protocol allows a client to retrieve bit xi, while keeping the value of the index i secret from the server. PIR can also retrieve blocks of data (e.g., an 'l' bit record) by viewing the database as 'n / l' elements, each with size 'l' bits.

Sultan.M, Department of Computer Science and Engineering, St. Michael College of Engineering and Technology, Sivagangai, India.

Karthi.M, Department of Computer Science and Engineering, Velammal College of Engineering and Technology, Madurai, India.

ManikandaPrabhu.M, Department of Computer Science and Engineering, St. Michael College of Engineering and Technology, Sivagangai, India.

Muruganandham.N, Department of Computer Science and Engineering, St. Michael College of Engineering and Technology, Sivagangai, India.

Tashelat Masleena.S, Department of Computer Science and Engineering, St. Michael College of Engineering and Technology, Sivagangai, India.

The ability to maintain privacy in information retrieval (hiding requesting user information from the server and other clients) has been a major application of private cloud. In the context of location-based services, PIR may be utilized to hide the location of a user.

Single-Server PIR is a family of protocols that assume a non - replicated database stored at a single site. These protocols, also known as computational PIR (cPIR), utilize certain cryptographic assumptions to ensure privacy. The protocols, which utilize certain cryptographic assumptions to ensure privacy, are known computational PIR. Initially, the client constructs a query Qi that is based on the index of the bit to be retrieved using a randomized query generation algorithm. The random query generator takes a description of the queries to run from a grammar file, which is similar to a YACC grammar. Then based on a reply generation algorithm at the server side, a response is generated. The reply generator works on the DB and returns an answer Ri to the client. The cost of the algorithm is high as it processes every bit in the query at the server side. This is because, if a single bit in the query is omitted, then the server can identify that the client is not interested in that particular information. This will reduce the privacy of the query. In most cases, the cryptographic algorithms to maintain the privacy of the query contain some sort of arithmetic operations. The cost of the evaluating the computational query in server side is quite high for large databases. The other option of sending the entire databases to the client side lapses security and requires high bandwidth and resource allocation from the server side. This will be a high factor of concern in wireless devices as the user is charged for downloading unwanted information from the database.

To overcome of the aforesaid issues, pCloud (Private Cloud) is introduced. pCloud is a distributed system that leverages the computational resources inside a peer-to-peer (P2P) cloud to speed up the processing of cPIR queries. A private cloud, also called an "internal cloud" or "corporate cloud," resides within the company environment (firewall) and its access is restricted, usually to company employees or business partners.

Publicly accessible databases are an indispensable resource for retrieving up-to-date information. But they also pose a significant risk to the privacy of the user, since a curious database operator can follow the user's queries and infer what the user is after. Indeed, in cases where the users' intentions are to be kept secret, users are often cautious about accessing the database. It can be shown that when accessing a single database, to completely guarantee the privacy of the user, the whole database should be downloaded, namely n bits should be communicated. A more efficient private information retrieval is designed by replicating the database. Distributed schemes that enable a user to access k replicated copies of a database and privately retrieve information stored in the database. This means that each individual server, holding a replicated copy of the database gets no information on the identity of the item retrieved by the user.

Private Information Retrieval (PIR) is a protocol that allows a client to retrieve an element of a database without the owner of that database being able to determine which element was selected. While this problem admits a trivial solution sending the entire database to the client allows the client to query with perfect privacy. Even though there exist techniques to reduce the communication complexity of this problem, but which will be nP-Hard for large databases. PIR is the only possible protocol that gives the user information theoretic privacy for their query in a single-server setting.

The problem of Private information retrieval (PIR) combines two major needs in the computer world. One is to access electronically and remotely information that is located in appropriately organized data bases. The second is to access these databases privately. Privacy is an increasing concern in open networks like the Internet.

Private information retrieval schemes enable a user to access a single or replicated copies of databases and retrieve information without revealing the identity of the sought items to any individual database. A major drawback of all PIR schemes known today is the assumption that the user knows the physical address of the sought item. This is usually not the case in most databases in use. Instead, the user typically holds a keyword (for example, the name of a specific company trade, the stock market), and the database internally converts this keyword to physical addresses. Information-theoretic privacy (or perfect privacy) is defined as the distribution of the queries the user sends to any server is independent of the index he wishes to retrieve. This means that each server cannot gain any information about user's interest regardless of his computational power.

Computational privacy is defined as the distributions of the queries the user sends to any server are computationally indistinguishable by varying the index. The general framework for computational private information retrieval is shown in figure 1. This means that each server cannot gain any information about user's interest provided that he is computationally bounded.

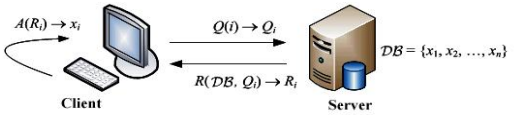

Figure 1: cPIR Framework

2. Related Work

A user wishes to privately retrieve the i-th bit in the database, without revealing any information about i. A distributed setting was suggested in which There are k, k >1, databases which hold copies of the same string x. The databases are allowed to communicate with the user, but not with one another. Previously several schemas were presented; all these schemas reflect the privacy of the information in the theoretic sense. But, later this is replaced with computational privacy. It is assumed that the databases are computationally bounded. Under appropriate intractability assumptions, the databases cannot gain information about i. The advantages of these schemes have significantly lower communication complexity than theoretic PIR's.

The cPIR class of algorithms does not rely on assumptions about non colluding servers. Instead, it is based on single-server architecture and employs well-known cryptographic primitives that guarantee query privacy in the presence of a computationally bounded server. The first cPIR protocol [18] relies on the quadratic residuosity assumption, which states that it is computationally hard to distinguish the quadratic residues in modulo arithmetic of a large composite modulus. Based on the above assumption, one may construct a cPIR protocol with a communication complexity , where is an arbitrarily small positive constant.

In [3] the first single-server protocol with poly-logarithmic communication complexity is introduced. The scheme builds upon the hiding assumption: it is hard to distinguish which of two primes divides for a hard-to-factor composite modulus m. The communication complexity of the protocol is, where 'a' depends on the desired security. The above asymptotic complexity is improved by [19] by introducing the advantages of length-flexible additively homomorphic public-key crypto-systems. It also allows the client to retrieve an l-bit block with a single answer, instead of a single bit that is common in most cPIR protocols.
The main limitation of the aforementioned cPIR protocols is their high computational cost because they require (for a single query) Ω(m) modular multiplications over a large modulus.

Distributed Db Architecture for Cpir Queries
The scalability problem of cPIR is focused in the proposed method, which involves designing a Distributed DB Architecture with peers for processing cPIR Queries.

A. Introduction

Cloud computing provides people the way to share distributed resources and services that belong to different organizations or sites. Cloud computing share the resources in open environment which creates security problem in cloud computing application. Introducing Privacy to pCloud which reduces response time of server while handling multiple clients at the same time in cloud and provide more security based on the client's requirement.

cPIR Algorithm

Setup (server)
1. e = CRT(DB)

Query Generation (client)
2. (P_1, P_2) = getPrimes(Π_i), m =$P_1 . P_2$
3. (g, |<g>|) = getGenerator(m), store h = $g^{|<g>|/\Pi_i}$
4. Send (m, g) to the server

Reply Generation (server)
5. Compute $g_e = g^e$ and return it to the client

Answer Extraction (client)
6. B_i = Pohlig-Hellman(h, $g_e^{|<g>|/\Pi i}$)

B. Cpir Protocol for Large Block Retrieval

An extended version of PIR protocol is proposed here which provides the necessary mathematical background and security considerations. A serious of implementation challenges, concerning the retrieval of blocks larger than 32 bytes. To tackle this challenge, a striping technique is introduced that allows PIR to efficiently retrieve a database page of arbitrary length, while maintaining the computational cost at the server constant.

PIR Protocol consists of four broad steps
1. Server Setup
2. Query Generation
3. Reply Generation
4. Answer Extraction

1. Server setup:
Let DB denote the database stored at the server and its size is n bytes. In the First Step, the Server segments the DB into 't' blocks, each of size 'l' bytes. Every block Bj (j=1,...,t) is associated with a prime power $\pi_j = p_j^{c_j}$ where $p_j \le 2^{8.l}$ is a small prime and $c_j = \lfloor 8.l / \log p_j \rfloor$. Observe that π_j has at least the same size (l) as Bj. The Value of smallest positive integer e is calculated e = Bj (mod π_j), for all j = 1,...,t, using Chinese Remainder Theorem (CRT). The server pre- computes e prior to receiving any queries for the current instance of DB.

2. Query generation by client:
In this step, the client generates and transmits its query to the server. Suppose that the client wishes to privately retrieve data residing in block Bi. It first computes two equal-length prime numbers P_1 and P_2, such that $P_1 = 2q_1\pi_i + P_2$ and $P_2 = 2q2d + 1$, where q1 and q2 are random primes, and d is a random number. Subsequently, it computes m = P1 * P2.

Next using a random element 'g' it generates a cyclic group <g> with order $|<g>| = q.\pi_i$, where q is an

$$g_e^q = g^{e|<g>|/\pi_i} = g^{B_i|<g>|/\pi_i}\, g^{E|<g>|} = g^{B_i|<g>|/\pi_i} = h^{B_i}$$

integer (i.e., g's order is divisible by π_i). It also keeps $q = |<g>| / \pi_i$ secret and stores $h = g^q$. The client sends query *Q = (m, g)* to the server and the Query Generation step concludes.

3. Reply generation:
During the third phase, the server evaluates $g_e = g^e$ and sends the result back to the client. According to the CRT, e can be written as $e = B_i + \pi_i . E$, for some E belongs to the group as shown in equation (1). This step processes the queries in parallel by assigning the queries to individual peers and returns the result. The information about the query is hidden from the

server. Even though, if any server comes to know the group of clients requesting the information bit, it can't be able to identify the particular client since the queries are distributed and processed in parallel with a private search key 'h'.

4. Answer extraction by the client:
In the last and fourth step of the protocol, the client can retrieve Bi by computing $\log_h g_e^q$ using the Pohlig-Hellman algorithm. The latter efficiently calculates the discrete logarithm within the subgroup H $\subset$ <g> of order $\pi_i = p_i^{c_i}$, when p_i is small. The Pohlig –Hellman algorithm is special purpose algorithm for computing discrete logarithms in a multiplicative group whose order is a smooth integer.

Striping Technique
The objective is to utilize cPIR in order to build a simple interface for efficiently retrieving pages of arbitrary size. A Black box mechanism for processing private query on indexed data is proposed here. In particular, we aim at providing a simple primitive *pGet (DB, i)* that privately retrieves the ith page from a database DB. The block size gravely impacts the performance of the protocol, even for values as small as 64 bytes. So striping technique is introduced as a solution to this problem.

Disk striping is a technique for spreading data over multiple disk drives. Disk striping can speed up operations that retrieve data from disk storage. The computer system breaks a body of data into units and spreads these units across the available disks. Systems that implement disk striping generally allow the user to select the data unit size or *stripe width.*

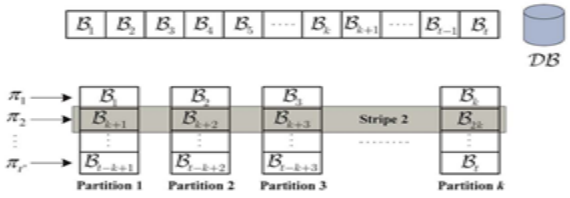

Figure 2: Striping Technique

Disk striping is available in two types. *Single user striping* uses relatively large data units, and improves performance on a single-user workstation by allowing parallel transfers from different disks. *Multi-user* uses smaller data units and improves performance in a multi-user environment by allowing simultaneous or overlapping read operations on multiple disk drives. Disk striping stores each data unit in only one place and does not offer protection from disk failure as shown in figure 2.

3. System Architecture

The detailed pCloud architecture is presented here. The goal of our system is to control the computational resources of co-operative peers, in order to accelerate the processing of cPIR queries. The pCloud organizes the peers in an overlay network, partitions the database into disjoint data segments, and disseminates the individual segments to the peers, in order to allow efficient query execution in parallel.

Server splits data to multiple peers to provide parallism to the clients. If the user want to retrieve data from server then a query needs to be generated to retieve block of data from server. Server will search for the peer in which the query is present using a private search key. The requested peer will send block of data to server. After receiving block of data the client will extract the data using polyhellman algorithm.

Two-tier architecture is implemented to process the cPIR's while maintaining the computational cost across multiple peers. This architecture slightly differs from existing P2P systems, which consists of two tiers as shown in figure 5. The first tier consists of the participating peers, which form an unstructured network overlay, where each node is connected to a number of random neighbors. The peers hold disjoint partitions of the database DB and are ready to answer queries upon request. The second tier is the database server, which holds the most current view of the entire DB. The inclusion of the server is necessary in our setting for the following reasons: 1) The PIR query needs to process every bit of the database. Suppose that the client does not receive replies for the entire different database partitions. This suggests that at least one partition has not been processed because either it did not exist in the network or the node that accommodated it failed 2) The server is the only entity that receives the data updates, and therefore, holds the most up-to-date DB instance at all times.

Database in server is splited among multiple peers using striping technique. Single server handling multiple peers takes more time to responds to the

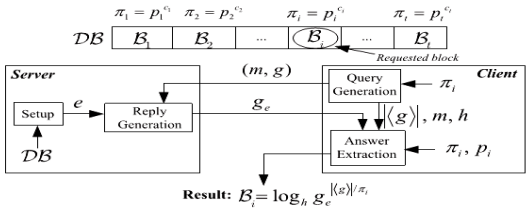

Figure 3: Interaction between client and server

clients. peers provides parallel execution so we can reduce response time of server considerably. If the user wants to retrieve data then the user should enter which block needs to be retrieved.Then query is generated to retrieve block of data from server. Server response to the client with a block of information which is generated based on user query. After receiving result from server client can extract the particular data from server. Integrity of received data from server is verified based on digital signature. In network, hackers can receive and modify the data in middle.

The actual challenge lies in how to partition and distribute DB to the peers, so that query processing experiences a linear performance speedup with respect to the number of partitions. If we simply subdivide DB into k partitions, and disseminate them among the peers, the query execution cost will be k times lower, compared to the traditional client/server model. Clearly, the size of the stripe (i.e.,k) adjusts a tradeoff between the computational cost at the peers and the communication cost at the client. On one hand, a large stripe size implies smaller partitions, i.e., better computational cost at the peers. On the other hand, if the stripe is large the client receives many redundant results since it is only interested in a single page within the stripe.

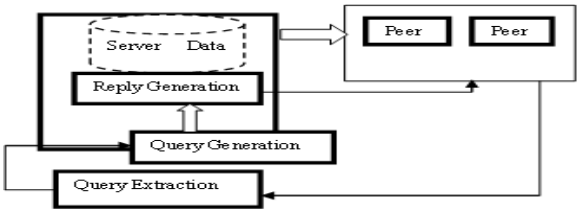

Fig.4: Architecture of the Query Generation – Response cPIR System

Analysis of Cpir Queries

The dependency of size of DB block 'n' becomes apparent when considering 10 MB vs. 1 MB databases. It can be seen that for 10 MB databases, cPIR become profitable if the bandwidth is below 210 Kbps, whereas in the case of a 1 MB database, this threshold goes down to 70Kbps. It is also important to note that for $\sqrt{n} < |N|$ this cPIR protocol would require more communication than a trivial database transfer. If the communication overheads are considered, the bandwidth thresholds below which cPIR becomes usable further decrease.

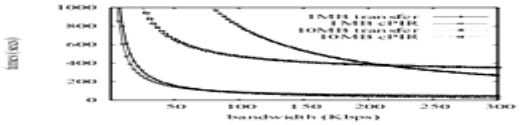

Figure.5: Low Bandwidth behavior of execution times for cPIR for the Pentium(R) 4 CPU set-ting vs. database transfer times

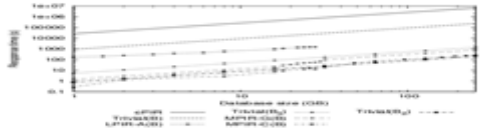

Figure.6: Comparing the response times of PIR schemes (cPIR), (LPIR-A), (MPIR-C), and (MPIR-G), as well as the trivial PIR scheme over different network bandwidths data using different database Sizes

4. Conclusion and Future Work

PCloud solution embeds a state-of-the-art PIR protocol in a distributed environment, by utilizing a novel striping technique. PCloud can retrieve arbitrarily large blocks of information with a single query. We present a comprehensive solution that includes a data placement policy, result retrieval, and authentication mechanisms. Specifically, compared to the traditional client/server architecture, pCloud drops the query response time by orders of magnitude, and its performance improves linearly with the number of peers. Future Enhancement is to can increase security of authentication more and more by using fingerprint technique and can verify integrity of data by using digital signature.

We conclude that many real-world situations that require privacy protection can obtain some insight from our work in deciding whether to use existing PIR schemes or the trivial download solution, based on their computing and networking constraints.

References

[1] Cachin, S. Micali, and M. Stadler, 1999 "Computationally Private Information Retrieval with Polylogarithmic Communication," Proc. Int'l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT '99).
[2] Chor, O. Goldreich, E. Kushilevitz, and M. Sudan,1995 "Private Information Retrieval," Proc. Symp. Foundations of Computer Science (FOCS '95).
[3] Coppersmith, 1996 "Finding a Small Root of a Univariate Modular Exponentiation," Proc. Int'l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT '96).
[4] R. Dingledine, N. Mathewson, and P.F. Syverson, 2004 "Tor: The Second-Generation Onion Router," Proc. Conf. USENIX Security Symp. (SSYM '04).
[5] W. Gasarch and A. Yerukhimovich, 2006 "Computationally Inexpensive cPIR," Unpublished Draft.
[6] W. Gasarch, 2004 "A Survey on Private Information Retrieval (Column: Computational Complexity)," Bull. of the European Assoc. for Theoretical Computer Science (EATCS), vol. 82, pp. 72-107.
[7] Gentry and Z. Ramzan, 2005 "Single-Database Private Information Retrieval with Constant Communication Rate," Proc. Int'l Colloquium on Automata, Languages, and Programming (ICALP '05).
[8] Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, 2008"Private Queries in Location Based Services: Anonymizers Are Not Necessary", Proc. ACM SIGMOD '08.
[9] Iliev and S.W. Smith, 2004 "Private Information Storage with Logarithm-Space Secure Hardware", Proc. Workshop Information Security, Management, Education, and Privacy.
[10] Khoshgozaran, H. Shirani-Mehr, and C. Shahabi, 2008 "SPIRAL: A Scalable Private Information Retrieval Approach to Location Privacy", Proc. Workshop Privacy-Aware Location-Based Mobile Services (PALMS '08).
[11] S. Yekhanin, "Towards 3-Query Locally Decodable Codes of Subexponential Length", J. ACM, vol. 55, no. 1, pp. 1-16, 2008.
[12] P. Williams and R. Sion, "Usable PIR", Proc. Symp. Network and Distributed System Security (NDSS '08), 2008.
[13] I. Stoica, R. Morris, D.R. Karger, M.F. Kaashoek, and H.Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications", Proc. ACM SIGCOMM '01, 2001.
[14] R. Sion and B. Carbunar, "On the Computational Practicality of Private Information Retrieval," Proc. Symp. Network and Distributed System Security (NDSS '07), 2007.

M.Sultan is currently a PG scholar in Department of Computer Science and Engineering at St.Michael.College of Engineering and Technology, Kalayarkoil, Sivagangai. He received his Bachelor Degree in Information Technology and Communication Engineering from K.L.N.College of Information Technology, Sivagangai, in 2009. His Research areas include grid computing, cloud computing and wireless sensor network security.

M.Karthi is currently a PG scholar in Department of Computer Science and Engineering at Velammal College of Engineering and Technology, Madurai. He received his Bachelor Degree in Information Technology and Communication Engineering from Sri Sowdambika College of Engineering, Virdhunagar District, Aruppukottai, in 2009. His Research areas include data mining and data warehousing, cloud computing and distributed system.

M.ManikandaPrabhu is currently a PG Scholar in the Department of Computer Science and Engineering at St.Michael College of Engineering and Technology, Sivagangai. He received his Bachelor Degree in Computer Science and Engineering from P.T.R College of Engineering and Technology, Madurai, in 2009. His Research areas include cloud computing, distributed system and network Security.

N.Muruganandham is currently a PG Scholar in the Department of Computer Science and Engineering at St.Michael College of Engineering and Technology, Sivagangai. He received his Bachelor degree in Computer Science and Engineering from Shanmuganathan Engineering College, Pudukottai, in 2010. His Research areas include cloud computing, grid computing, and wireless sensor network security.

S.Tashelat Masleena is currently a PG Scholar in the Department of Computer Science and Engineering at St.Michael College of Engineering and Technology, Sivagangai. She received his Bachelor degree in Computer Science and Engineering from Mohamed Sathak Engineering College, Keelakarai district, Ramanathapuram, in His Research areas include data mining and data warehousing, cloud computing and wireless sensor network security.