IFERP

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 7, July 2016

Traffic Redundancy Elimination for Cost and Bandwidth Reduction in Cloud Using Prediction Acknowledgement (PACK)

[1] Ashwini A. Ghorpade, [2] S.R.Idate
[1] M.Tech, [2] Associate Professor,
[1][2]Department of Information Technology,
Bharati Vidyapeeth Deemed University College of Engineering, Pune, India
[1]arnawadkar@gmail.com, [2] sridate@bvuceoe.edu.in

***Abstract-* — PACK (Predictive ACKs), is a novel end-to-end traffic redundancy elimination (TRE) system. PACKs main advantage is its capability of offloading the cloud-server TRE e ort to end-clients, thus minimizing the processing costs induced by the TRE algorithm. PACK is based on a novel TRE technique, which allows the client to use newly received chunks to identify previously received chunk chains, which in turn can be used as reliable predictors to future transmitted chunks. The main aim is to eliminate redundant data transfer and reduce bandwidth cost on the network, in particular traffic redundancy elimination (TRE), for reducing bandwidth costs.**

***Index Terms*— Chunking mechanism, Cloud Computing; Traffic Redundancy System**

I. INTRODUCTION

Cloud computing represents the latest step in a half century of computing technology evolution, beginning with the mainframe and then client-server, PC, Internet, and mobility. The redundant traffic in network uses more network bandwidth hence reducing the efficiency of the internet. The traffic redundancy elimination technique can be used to remove the data redundancy within random network flows; it can be considered as effectual technique of upgrading the efficiency of the network links. In TRE, it is used to check whether the file is already present or not in network link flows. If a more copies of file are found in network flow means it can remove from that and keep it original file then it passes to the cloud data storage to save that file. Eliminating redundant network traffic became an important task with the volume increase of large files and rich media content. Consequently, many commercial traffic redundancy elimination (TRE) middleboxes, were placed at WAN and Internet access points.

There are three major parts Client, Application Server and Cloud. Client role is to upload their file; it may be single or multiple le on the cloud. Application server divides these files into chunks of 64 bits each and store on cloud side for checking duplicate chunk on next time. If client upload a new file application server first divide it into number of chunks and check that chunk with previously stored chunk, if matched then it can't send data on cloud side, the available duplicate chunks replace with it.

Server cloud role is to accumulate the chunks that are provided by application server, and store file uploaded by clients. Using the abiding chunk's metadata statistics kept locally, the receiver sends to the server prediction which includes chunk's signatures and verification hints of the sender's forthcoming data. The sender first scrutinizes the hint and accomplishes the TRE operation only on a hint-match. The persistence of this technique is to escape the exclusive TRE computation at the sender side in the absence of traffic redundancy. When redundancy is distinguished, the sender then ends to the receiver only the ACKs to the predictions, as a replacement for of sending the data.

II. RELATED WORK

1] WANAX

WANAX [6] is a wide area network accelerator designed for reducing traffic in WAN. It uses a novel multi resolution chunking (MRC) scheme that encompasses high compression rates as well as high disk performance for a range of content by means of using much less memory than other open approaches. WANAX utilize the design of MRC to perform intelligent load shedding to exploit throughput when consecutively running on resource-limited shared platforms. WANAX make use of the mesh network environments being set up in the increasing world, as an alternative of just using the star topologies regular in enterprises. The large amount of work is done for this system for optimization.

The chunking scheme used in WANAX is multi resolution scheme i.e. MRC .MRC joins the rewards of both large and small chunks by permitting multiple chunk sizes to conjugate subsist in the system. WANAX uses MRC to achieve, High compression rate, low disk seek and low memory demands. When content overlap is maximum, WANAX can utilize bigger chunks to decrease disk seeks and memory demands. Conversely, when larger chunks ignore compression opportunities, WANAX uses smaller chunk sizes to get higher compression. In disparity, existing WAN accelerators usually use a fixed chunk size, which is termed as *single-resolution chunking*, or SRC.

Some drawbacks consist in WANAX are End-to-end traffic is not handled by middle boxes as it is encrypted. It generates latency for non cached data and middle boxes will not advance the performance

2] A Low-Bandwidth Network File System

Low-Bandwidth Network File System is a network file system designed for low-bandwidth network system. LBFS utilizes resemblance between files or versions of the same file to set aside bandwidth. It evades sending data over the network when the same data can already be there in the server's file system or the client's cache. By means of this technique in combination with conventional compression and caching, LBFS uses over an order of magnitude less bandwidth than traditional network file systems on ordinary workloads.

LBFS [4] is designed to save bandwidth at the same time providing traditional file system semantics. Particularly, LBFS provides close-to-open consistency. After a client completes write operation and closed a file, a new client opening the same file will constantly see the fresh contents. Additionally, once a file is profitably written and closed, the data is inherent in securely at the server. To save bandwidth, LBFS uses a outsized, persistent file cache at the client [5]. LBFS presume clients will have sufficient cache to hold a user's complete working set of files with such antagonistic caching; most client–server communication is exclusively for the purpose of preserving consistency.

At the both client and server side, LBFS must index a set of files to distinguish between data chunks it can evade sending over the network. To keep chunk transfers, LBFS relies on the anti-collision properties of the SHA-1 hash function. The possibility of two inputs to SHA-1 producing the same output is far lesser than the possibility of hardware bit errors. As a result, LBFS pursue the broadly acknowledged practice of presuming no hash conflict. If the client and server both have data chunks constructing the same SHA-1 hash, they presume the two are actually the same chunk and evade relocating its contents over the network.

Lacunae in the LBFS systems are LBFS can be used only for short bandwidth network file system. It cannot be used for any other type of environment. For TRE operation data must be modified.

3] End-RE

End-RE [2] is an alternate approach where redundancy elimination (RE) is provided as an end system service. Unlike middleboxes, such an approach remunerates both end-to-end encrypted traffic as well as traffic on last-hop wireless links to mobile devices

End-RE is designed to optimize data transfers in the direction from servers in a remote data center to clients in the organization, since this captures majority traffic. For easy deployment, the End-RE service should necessitate no modification to existing applications run within clients through which we can acquire transparency in the system. For fine grained operation and to advance the end-to-end latencies and provide bandwidth savings for short flows, End-RE must function at fine granularities, restraining duplicate byte strings as small as 32-64B. As working on fine granularities can assist recognizing better amounts of redundancy, it can also oblige considerable computation and decoding overhead, making the system not viable for devices like cell phones. End-RE is designed to opportunistically control CPU resources on end hosts when they are not being used by other applications. End-RE must adjust its use of CPU based on server load. This ensures Fast and adaptive encoding at server side End-RE depends on data caches to perform RE. However, memory on servers and clients could be partial and may be dynamically used by other applications. Therefore, End-RE must use as minimum memory on end hosts as feasible through the use of optimized data structures.

Fingerprinting is the chunking mechanism used in End-RE various fingerprinting techniques are used in End-RE such as MAXP, MODP, FIXED, SAMPLEBYTE**.** MAXP and MODP are content-based and a thus robust to small change in content, while FIXED is content agnostic but computationally efficient therefore SAMPLEBYTE fingerprinting is used to combine the robustness of a content-based approach with the computational efficiency of FIXED. [3]

There are some drawbacks for this system first is it is server specific redundancy elimination technique. And Chunk size is small in case of End-RE

4] Novel TRE (Predictive Acknowledgement)

PACK (Predictive Acknowledgement) is a novel end-to-end traffic redundancy elimination (TRE) system, deliberated for users who use cloud computing technology. Cloud-based TRE requires being concern with a well refereed use of cloud resources so that the bandwidth and cost reduction merge with the added cost of TRE computation and storage of data would be optimized. PACK's main advantage is its proficiency of to hand over the cloud- server TRE attempt to end-clients, thus dropping the processing cost sustained by the TRE algorithm. A variety of previous solutions, PACK doesn't need the server to continuously sustain the clients' status. This makes PACK very suitable for pervasive computation surroundings that unite client mobility and server relocation to keep up cloud flexibility.
PACK is based on a novel TRE technique, which allows the client to use recently received chunks to identify previously received chunk chains, which in turn can be used as consistent predictors to future transmitted chunks.

III. PROPOSED WORK

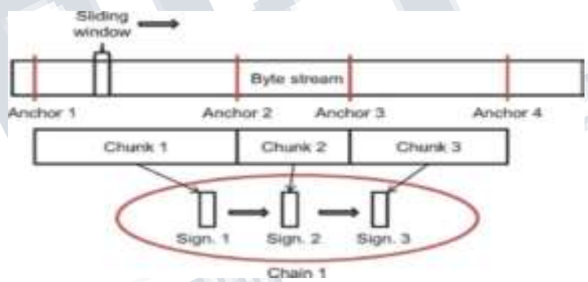
A. Design Considerations:

- The system is used for lay off the broadcasting of monotonous and redundant content.
- It uses influence of forecasts to neglect redundancy of traffic in the cloud and its end-users.
- It regulates the reduction of cost which is accomplished by cloud while attaining the additional bandwidth savings at the client side.
- The new computationally lightweight chunking scheme is attained.

B. Description of the Proposed Algorithm:

- ***Pack Algorithm***

The representation of chunking mechanism of PACK algorithm is shown in the figure.



C. Receiver Chunk Store

PACK uses a new chains method, as shown in above figure, in which chunks are linked with other chunks as per to their previous received order. The PACK receiver keeps a chunk store, which is a huge size cache of chunks and their linked metadata. Chunk's metadata comprises the chunk's signature and a solo pointer to the consecutive chunk in the last received stream includes this chunk. Caching and indexing methods are deployed to proficiently preserve and recover the stored chunks, signatures, and the chains created by navigating the chunk pointers.

D. Receiver Algorithm

When the new data is arrived the receiver computes the individual signature for each chunk and check for a match in the local chunk store. If the chunk's signature is found, the receiver decides whether it is a part of a previously received chain, using the chunks' metadata. If it matches, the receiver sends a prediction to the sender for some subsequently predictable chain chunks. The prediction holds a starting point in the byte stream i.e. offset and the identity of several succeeding chunks i.e. PRED command.

E. Sender Algorithm

When a sender gets a PRED message from the receiver, it tries to match the received predictions to its buffered data. For every prediction, the sender decides the equivalent TCP sequence range and validates the hint. After a hint match, the sender computes the more computationally rigorous MD-5 signature for the predicted data range it also evaluate the result to the signature received in the PRED message. If the hint doesn't match, a computationally extensive operation is saved. If the two MD-5 signatures matches, the sender can securely take for granted that the receiver's prediction is exact. In this case, it restores the corresponding outgoing buffered data with a PRED-ACK message.

IV. PSEUDO CODE

1] Module 1: Login Authentication and Database Designing

This module contains login module which authenticates user and implemented on server side. It also contains facility to upload and download a le. Creating database at Server side, we receive the request from client and send the response to respected request.

2] Module 2: Creating chunks and Make Prediction

This module consists of PACK algorithm implementation. Generating chunks at client side and it will send to server. Server makes prediction for receiving chunks.

3] Module 3: Sender Algorithm

This module consist the implementation of Sender Algorithm in the client side. On receiving data it create a unique key using MD-5 Algorithm and find respective key in chunk store. If generated key matched in chunk store then it sent prediction with TCP ACK else, it sent TCP ACK only.

4] Module 4: Receiver Algorithm and Deployment

This module consist the implementation of server side Receiver Algorithm. On receiving prediction from server with TCP ACK, then it will generate the key for chunk with MD-5 Algorithm which is prepared to send and compare with received prediction. If prediction is right then it will sent PRED ACK to server, else chunk will sent.

IV. SIMULATION RESULT

System	Average Size of Chunk	Sender-side chunking system	Sender signing	Receiver-side chunking system	Receiver Signing method
WANAX [6]	Several	Multi-Resolution Chunking (MRC) of all data	SHA-1: all data	Multi-Resolution Chunking (MRC) of all data	SHA-1: real data
LBFS	Flexible, 8KB	Rabin: modified data	Rabin: modified data	Rabin: all modified data	Rabin: all modified data
End-RE	Limited, 32-64 bytes	Sample Byte: all data	SHA-1: all data	Not any	Not any
PACK (Novel TRE)	Unlimited, Depend on receiver's choice	Not any	SHA-1: true predictions and 0.4% of false predictions	PACK chunking real data chunking	SHA-1: real data
TRE system with PACK i.e. i.e. Proposed application	Depend upon the receiver choice	No	Message digest 5	PACK Real data chunking	MD 5 creation of a 128-bit message digest from data input

Table 1: comparison of system with other TRE techniques

Figure 1: login form

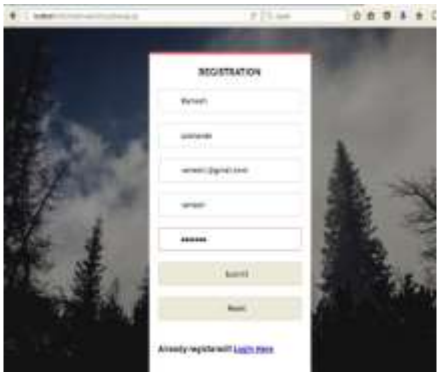


Figure 2: registration form

Figure 3: form for uploading a file

Figure 4: form for downloading a file

V. CONCLUSION

This application is designed for cloud computing environment. It is an innovative receiver-side traffic redundancy elimination way out that is influenced by the power of predictions to neglect the unnecessary traffic between the cloud and its end users. This system provides solution for the problem that every single receiver analyzes the incoming data and attempts to match the chunks with the earlier received chunk chain of a local file. As the middle-box and other existing solutions are insufficient for the redundancy requirements the PACK traffic redundancy elimination system redefines the cloud technology. From the conclusions drawn two remarkable future developments can deliver advances to the Predictive Acknowledgement concept. First are the implementation bear chains by analyzing only the last observed consequent chunk in a least recently used (LRU) methodology. The second future expansion is the optimization techniques of the hybrid sender–receiver methodology that are reliant on the shared data. This result in server’s cost changes.

REFERENCES

1) G.Krishnaveni, D.SriLakshmi Advanced Prediction-Based System for Cloud Band-width and Cost Reduction IJCST Vol.4,Issue Spl-4,Oct-Dec 2013.

2) E.Zohar, I.Cidon, O.Mokryn;”PACK: Prediction-Based Cloud Bandwidth and Cost Reduction System”; IEEE/ACM Transactions on Networking; 2013; 1063-6692 IEEE

3) C.Muthukrishnan, A. Anand, A. Akella, R. Ramjee. “Redundancy in Network Traffic: Finding and Implications” Proc. SIGMETRICS, 2009.

4) S. Ihm, K Park, V. Pai, “Wide area network acceleration for the developing world”, in proc. USENIX ATC, 2010,pp 18-18

5) A. Muthitacharoen, B. Chen, D. Mazières, “A low-bandwidth network file system,” in Proc. SOSP, 2001, pp. 174–187.

6) B.Aggarwal, A.Akella, A.Anand, A. Balachandran, P.Chitnis, G. Varghese C. Muthukrishnan, R. Ramjee , ”End-RE: An end-system redundancy elimination service for enterprises,” in Proc. NSDI,2010.

7) U. Manber, “Finding similar files in a large file system,” Proc. USENIX Winter Tech. Conf., 1994, pp. 1–10.

8) M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van derMerwe. Design and implementation of RCP.In NSDI, 2005.

9) A. Davie and Y. Rekhter. MPLS: technology and applications. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

10) U. Erlingsson, M. Manasse, and F. McSherry. A cool and practical alternative totraditionalhashtables.InWDAS, 2006.

11) Scott Shenker. "Packet caches on routers",Proceedings of the ACM SIGCOMM 2008,Conference on Data communication - SIGCOMM 08 SIGCOMM 08, 2008.

12) Yan Zhang, , and Nirwan Ansari. "On Protocol-Independent Data Redundancy Elimination",IEEE Communications Surveys & Tutorials, 2014.

13) A.Ghorpade, S.Idate. "Survey of Traffic Redundancy Elimination systems for cloud computing environment to Reduce Cloud Bandwidth and Costs", Int.J.ComputerTechnology&Applications,Vol 7(2), 2016