IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-6, Special Issue-2, March 2018 E-ISSN: 2347-2693

Traffic Redundancy and Elimination Approach to Reduce Cloud Bandwidth and Costs

P.Joseph Charles[1*], S.Thulasi Bharathi[2], Angel Preethi[3]

[1] Department of information technology, St.Joseph's College, tirchy-02
[2] *Department of information technology, St.Joseph's College, tirchy-02*
[3] *Department of information technology, St.Joseph's College, tirchy-02*

**Corresponding Author: charles_pjm74@gmail.com*

Available online at: www.ijcseonline.org

Abstract— Cloud computing is expected to trigger high demand for Traffic Redundancy Elimination(TRE) solutions as the amount of data exchanged between the cloud and its users is expected to dramatically increase. We present PACK (Predictive ACKs),a novel end-to-end TRE system, designed for cloud computing customers .The cloud environment redefines the TRE system requirements, making proprietary middle-box solutions inadequate .To reduce bandwidth cost ,Cloud –Based Traffic redundancy elimination(TRE) system should make use of sophisticated use of cloud resources, so that the additional cost of TRE computation and storage can be optimized .This TRE technique uses Predictive. ACK's(PACK), designed for cloud computing customers. It gives a methodology to reduce the cloud bandwidth by making use of predictions for the future data, thereby eliminating redundant data. It is a receiver driven TRE technique, that allows the receiver to use newly received chunks to identify previously received chunk chains , that can be used to send predictions for the subsequent data.. This technique does not require the sender to continuously maintain the receiver's status, unlike traditional approach. Predictive ACK's is suitable in pervasive computation environment. It is transparent to all TCP based application and network devices .The main advantage of PACK is that it can offload cloud –server TRE effort to end client, thereby minimizing the processing cost induced by the TRE algorithm.

Keywords—Traffic redundancy elimination, cloud computing, predictions

I. INTRODUCTION

Cloud customers pay only for the actual use of computing resources, storage, and bandwidth, according to their changing needs, utilizing the cloud's scalable and elastic computational capabilities. In particular, data transfer costs (i.e., bandwidth) is an important issue when trying to minimize costs. Consequently, cloud customers, applying a judicious use of the cloud's resources, are motivated to use various traffic reduction techniques, in particular traffic redundancy elimination (TRE), for reducing bandwidth costs. Traffic redundancy stems from common end-users' activities, such as repeatedly accessing, downloading, uploading (i.e., backup), distributing, and modifying the same or similar information items (documents, data, Web, and video). TRE is used to eliminate the transmission of redundant content and, therefore, to significantly reduce the network cost. In most common TRE solutions, both the sender and the receiver examine and compare signatures of data chunks, parsed according to the data content, prior to their transmission. When redundant chunks are detected, the sender replaces the transmission of each redundant chunk with its strong signature. Commercial TRE solutions are popular at enterprise networks, and involve the deployment of two or more proprietary-protocol, state synchronized middle-boxes at both the intranet entry points of data centers and branch offices, eliminating repetitive traffic between them (e.g., Cisco, Riverbed, Quantum, Juniper, Blue Coat, Expand Networks, and F5). While proprietary middle-boxes are popular point solutions within enterprises, they are not as attractive in a cloud environment. Cloud providers cannot benefit from a technology whose goal is to reduce customer bandwidth bills, and thus are not likely to invest in one.

The rise of "on-demand" work spaces, meeting rooms, and work-from-home solutions detaches the workers from their offices. In such a dynamic work environment, fixed-point solutions that require a client-side and a server-side middle-box pair become ineffective. On the other hand, cloud-side elasticity motivates work distribution among servers and migration among data centers. Therefore, it is commonly agreed that a universal, software-based, end-to-end TRE is crucial in today's pervasive environment. This enables the use of a standard protocol stack and makes a TRE within end-to-end secured traffic (e.g., SSL) possible. Current end-to-end TRE solutions are sender-based. In the case where the cloud server is the sender, these solutions require that the server continuously maintain clients' status. It show here that cloud elasticity calls for a new TRE solution. First, cloud load balancing and power optimizations may lead to a server-side process and data migration environment, in which TRE solutions that require full synchronization between the server and the client are hard to accomplish or may lose efficiency due to lost synchronization. Second, the popularity of rich media that consume high bandwidth motivates content distribution network (CDN) solutions, in which the service point for fixed and mobile users may change dynamically according to the relative service point locations and loads. Clearly, a TRE solution that puts most of its computational effort on the cloud side may turn to be less cost-effective than the one that leverages the combined client-side capabilities. Given an end-to-end solution, it have found through our experiments that sender-based end-to-end TRE solutions add a considerable load to the servers, which may eradicate the cloud cost saving addressed by the TRE in the first place. This experiment further show that current end-to-end solutions also suffer from the requirement to maintain end-to-end synchronization that may result in degraded TRE efficiency. In this paper present a novel receiver-based end-to-end TRE solution that relies on the power of predictions to eliminate redundant traffic between the cloud and its end-users. In this solution, each receiver observes the incoming stream and tries to match its chunks with a previously received chunk chain or a chunk chain of a local file. Using the long-term chunks' meta-data information kept locally, the receiver sends to the server predictions that include chunks' signatures and easy-to-verify hints of the sender's future data. The sender first examines the hint and performs the TRE operation only on a hint-match. The purpose of this procedure is to avoid the expensive TRE computation at the sender side in the absence of traffic redundancy. When redundancy is detected, the sender then sends to the receiver only the ACKs to the predictions, instead of sending the data. On the receiver side, it proposed a new computationally lightweight chunking (fingerprinting) scheme termed PACK chunking. PACK chunking is a new alternative for Rabin fingerprinting traditionally used by RE applications. Experiments show that this approach can reach data processing speeds over 3 Gb/s, at least 20% faster than Rabin fingerprinting.

II. RELATED WORK

A TRE system called WANAX[1] was developed for the developing world where storage and WAN bandwidth are scarce. For expensive commercial hardware WANAX is a software – based middle-box replacement. The receiver middle box receives the data signatures from the sender middle box. The sender middle sends the signature holding back the TCP stream. The receiver checks whether the data chunks are found in the local cache. If the data chunks are not found in the local cache the data is obtained from the sender middle box or a nearby receiver middle-box. But such a scheme incurs a three way handshake latency for non cached data. ENDRE[2] is an end-to-end sender- based RE for enterprise networks. It uses a new chunking scheme that is faster than the commonly used Rabin fingerprint. The size of the chunks are as small as 32-64 B. The server has to maintains a full and reliable synchronized cache for each client. The size of the cache is kept small , hence the system is not suitable for medium-to-large content or long term redundancy.

III. EXISTING SYSTEM

Traffic redundancy stems from common end-users' activities, such as repeatedly accessing, downloading, uploading (i.e., backup), distributing, and modifying the same or similar information items (documents, data, Web, and video). TRE is used to eliminate the transmission of redundant content and, therefore, to significantly reduce the network cost. In most common TRE solutions, both the sender and the receiver examine and compare signatures of data chunks, parsed according to the data content, prior to their transmission. When redundant chunks are detected, the sender replaces the transmission of each redundant chunk with its strong signature. Commercial TRE solutions are popular at enterprise networks, and involve the deployment of two or more proprietary-protocol, state synchronized middle-boxes at both the intranet entry points of data centers.

A. DISADVANTAGES OF EXISTING SYSTEM

Cloud providers cannot benefit from a technology whose goal is to reduce customer bandwidth bills, and thus are not likely to invest in one.

The rise of "on-demand" work spaces, meeting rooms, and work-from-home solutions detaches the workers from their offices. In such a dynamic work environment, fixed-point solutions that require a client-side and a server-side middle-box pair become ineffective.cloud load balancing and power optimizations may lead to a server-sideprocess and data migration environment, in which TRE solutions that require full synchronization between the server and the client are hard to accomplish or may lose efficiency due to lost synchronization Current end-to-end solutions also suffer from the requirement to maintain end-to-end synchronization that may result in degraded TRE efficiency.

IV. PROPOSED SYSTEM

The proposed system is a receiver based TRE solution which makes use of predictions to eliminate redundant traffic between cloud and its users. The receiver receives the incoming data stream from the sender. The receiver parses the incoming stream into variable –size , and generates content based signed chunks using SHA-1.

The chunks are compared to the local storage called chunk store .If a matching is found the receiver retrieves a sequence of subsequent chunks called chain .The receiver sends a prediction to the sender for the subsequent data. The prediction include chunks signature and easy to verify hints of the senders future data. The sender verifies the hint and only on the hint match it performs the expensive SHA-1 operation . If redundancy is detected the sender send ACK to the prediction else it sends raw data. The computations are performed by a group of clients instead of cloud distributing the load as the client process only the TRE part. It supports the client to interact with multiple servers and the servers are dynamically relocated around the cloud .It reduces 20% of overall cost to the customer 30% redundancy elimination. It uses the TCP options field and hence all TCP based applications are supported.During the initial TCP handshake PACK is enabled on both the sides. This is done by adding a PACK permitted flag to the TCP options field .PACK assumes that the data is redundant. The sender sends the data in one or more TCP segments and the receiver checks whether the currently received chunk is similar to the chunk in the chunkstore .If the chunks are similar then the receiver sends a TCP ACK message and embeds the prediction in the options field of the packet. If the prediction is true the sender responds with PRED- ACK instead of actual data.

The stream of data received at the PACK receiver is parsed to a sequence of variable-size, content-based signed chunks. The chunks are then compared to the receiver local storage, termed chunk store. If a matching chunk is found in the local chunk store, the receiver retrieves the sequence of subsequent chunks, referred to as a chain, by traversing the sequence of LRU chunk pointers that are included in the chunks' metadata.

Using the constructed chain, the receiver sends a prediction to the sender for the subsequent data. Part of each chunk's prediction, termed a hint, is an easy-to-compute function with a small-enough false-positive value, such as the value of the last byte in the predicted data or a byte-wide XOR checksum of all or selected bytes. The prediction sent by the receiver includes the range of the predicted data, the hint, and the signature of the chunk. The sender identifies the predicted range in its buffered data and verifies the hint for that range. If the result matches the received hint, it continues to perform the more computationally intensive SHA-1 signature operation.Upon a signature match, the sender sends a confirmation message to the receiver, enabling it to copy the matched data from its local storage.

A. Sender algorithm

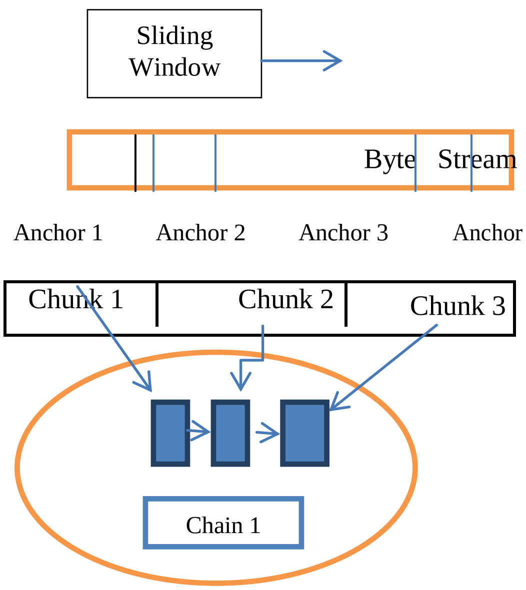
The sender receives the PRED message from the receiver. On receiving the prediction the sender tries to match the prediction with that of the buffered data. The sender determines the corresponding range for each prediction and verifies the hint .The prediction command consists of chunks signature, an easy to compute function called hint . Only on the hint match the sender calculates the signature using the SHA-1 for the predicted data range and matches it with the signature in the received prediction message. If a match is found then the sender sends an Acknowledgement for the predictions and replaces outgoing buffered data with PRED-ACK. If the predictions are false the sender continues with normal operation.

B. Receiver algorithm

The receiver receives the incoming stream and parses it into chunks. It computes the signature for each chunk and matches it with the local chunk store. If a match is found then the receiver retrieves the chunk from the chunk store and sends a prediction to the sender for subsequent next expected chunks. The prediction carries an offset, and identity of many subsequent chunks. If the prediction is successful the sender responds with PRED-ACK message . The receiver copies the corresponding data from chunk store to its TCP input buffers. Then the receiver sends a normal TCP ACK with the next expected TCP sequence number. If one or more predicted chunks are already sent, the sender resumes with normal operation.

C. Receiver chunk store

A new chain scheme is used by the PACK as shown in Fig 1. This will store the data in the form of chunks. The chunkstore consists of chunks that are linked together according to the last received order .It consists ofalargecacheofchunks andtheirassociated metadata. The metadata consists of chunks signature and a pointer to the successive chunk in the last received order. To retrieve the stored chunk, their signatures and the chain , caching and indexing techniques are employed.



V. CONCLUSION AND FUTURE WORK

This system overcomes the challenges faced by the proprietary middle box solution. PACK is a receiver based end- to – end TRE based on novel speculative principles that reduce latency and cloud operational costs .It is an efficient TRE technique which can eliminate the redundant data based on the power of predictions .Future work may include , if the changes in data are scattered PACK's receiver-based mode is less efficient. So ,when PACK identifies a pattern of scattered changes ,it should drive a sender- receiver approach depending on shared decisions derived from receiver's power or server's cost changes.

REFERENCES

[1] U. Manber, "Finding similar files in a large file system," in Proc. USENIX Winter Tech. Conf., 1994, pp. 1–10.

[2] N T. Spring and D. Wetherall, "A protocol-independent technique for eliminating redundant network traffic," inProc. SIGCOMM, 2000, vol.30, pp. 87–95.

[3] A. Muthitacharoen, B. Chen, andD. Mazières, "A low-bandwidth network file system," inProc. OSP, 2001,174–187.

[4] S. Ihm, K. Park, and V. Pai, "Wide-area network acceleration for the developing world," inProc. USENIX ATC, 2010, pp. 18–18

[5] B. Aggarwal, A. Akella, A. Anand, A. Balachandran, P. Chitnis, C. Muthukrishnan, R. Ramjee, and G. Varghese,

[6] E. Zohar, I. Cidon, and O. Mokryn, "The power of prediction: Cloud bandwidth and cost reduction," inProc. SIGCOMM, 2011, pp. 86–97.