Cost Reduction System through TRE Based on Prediction

Priyanka M. Gawande[1], A. A. Nikose[2]

Department of Computer Science and Engineering, Nagpur University Priyadarshini Bhagwati College of Engineering, Nagpur-India

Abstract: ***In this paper, we present Cost Reduction System through TRE based on prediction which is a receiver based end-to-end traffic redundancy elimination "TRE" system. This is designed for the cloud computing customers. The Cloud-based TRE system needs to apply a judicious use of cloud resources to combine the bandwidth cost reduction with the additional cost of TRE computation and the storage would be optimized. The main advantage of the Predictive Acknowledgement (PACK) is its capability of offloading the cloud-server TRE effort to end clients. This results in minimizing the processing costs which is induced by the TRE algorithm. Here, the PACK does not require the server to continuously maintain the clients' status unlike the previous solutions. This makes suitable for pervasive computation environments that combine the client mobility and server migration to maintain the cloud elasticity. The Cost Reduction System through TRE based on prediction is based on a prediction TRE technique, that allows the client to use the newly received chunks for identifying the previously received chunk chains, which in turn can be used as the reliable predictors to future transmitted chunks. We present a fully functional PACK implementation, which is transparent to all TCP-based applications and network devices. Finally, we analyze the Prediction Acknowledgement benefits for cloud users, using the traffic traces from various sources.***

Keywords: PACK, TRE, cloud-server, client mobility, TCP

1. Introduction

The main aim of this project is Prediction Acknowledgement (PACK) based novel TRE technique, which allows the client to use newly received chunks to identify the previously received chunk chains, which in turn can be used as reliable predictors for the future transmitted chunks. Cloud computing offers its customers an economical and convenient *pay-as-you-go* service model, known also as *usage-based pricing*. The Cloud customers pay only for the actual use of computing resources, storage and bandwidth cost according to their changing needs thus utilizing the cloud's scalalable and elastic computational capabilities. In particular, the data transfer costs (i.e., bandwidth) is an important issue when we are trying to minimize the costs. Most of the cloud customers, judiciously using the cloud's resourcces, are motivated to use various traffic reduction techniques, in particular traffic redundancy elimination (TRE), for reducing the bandwidth costs. The problem of Traffic redundancy stems from common end-users' activities, such as repeatedly accessing, downloading, uploading (i.e., backup), distributing, and modifying the same or similar type of information items (documents, data, Web, and video). TRE is a technique used to eliminate the transmission of redundant content and, therefore then significantly reduce the network cost. In most of the common TRE solutions, both the sender and the receiver examine and compare the signatures of data chunks, parsed according to the data content, prior to their transmission. When the redundant chunks are detected, the sender then replaces the transmission of each redundant chunk with its strong signature. The Commercial TRE solutions are popular at enterprise networks, which involves the deployment of two or more proprietary-protocol, state synchronized middle-boxes at both the intranet entry points of data centers and the branch offices, eliminating repetitive traffic between them. Here we 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.

2. Literature Survey

1] WANAX

Wanax as proposed in [6] is a wide area network accelerator anticipated for reducing traffic issues respecting with the Wide Area Network. It applies a novel multi resolution chunking (MRC) scheme that encompasses not only high compression rates but also high disk performance for a variety of content with the help of using much less memory than other various open approaches. Wanax make use of 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 demanding 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.

WANAX is motivated by the challenges of bandwidth issues occurred in the cloud computing environment. The design of this system is designed with respect to achieve maximum compression, minimum disk seek, minimum memory load and to exploit local resources

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.

EndRE 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 EndRE 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, EndRE 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. EndRE is designed to opportunistically control CPU resources on end hosts when they are not being used by other applications. EndRE must adjust its use of CPU based on server load. This ensures Fast and adaptive encoding at server side EndRE 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, EndRE 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

3. Existing System

The Cloud providers cannot be benefitted from a technology whose goal is to reduce the customer bandwidth bills, and thus are not likely to invest in one. The rise of the "on-demand" work spaces, meeting rooms, and work-from-home solutions detaches most of the workers from their offices. In such a dynamic work environment, the fixed-point solutions that require a client-side and a server - side middle - box pair becomes ineffective. The Cloud load balancing and power optimizations may lead to a server-side process and data migration environment, in which the TRE solutions that requires full synchronization between the server and the client are hard to accomplish. Thus it may lose efficiency due to lost synchronization. The Current end-to-end solutions also suffer from the requirement to maintain the end-to-end synchronization that may result in degraded TRE efficiency.

4. Implementation

In this paper, we are presenting a novel receiver-based end-to-end TRE solution that relies on the power of predictions to eliminate the redundant traffic between the cloud and its end-users. In this solution, the receiver first observes the incoming stream and tries to match its chunks with the previously received chunk chain or a chunk chain of a local file. Using the long-term chunks' metadata information kept locally, the receiver then sends to the server predictions. This predictions include the chunks' signatures and it becomes easy-to-verify the hints of the sender's future data.

On the receiver side, we propose a new computationally lightweight chunking scheme termed as *PACK chunking*. PACK chunking is a new alternative for Rabin fingerprinting traditionally used by RE applications. Our approach can reach the data processing speeds over 3 Gbps, at least 20% faster than Rabin fingerprinting. The receiver-based TRE solution addresses the mobility problems common to the quasi mobile desktop/ laptops computational environments. One of them is the cloud elasticity due to which the servers are dynamically relocated around the federated cloud, thus causing clients to interact with the multiple changing servers. We implemented, tested, and performed realistic experiments with the PACK within a cloud environment. Our experiments demonstrate a cloud cost reduction achieved at a reasonable client effort while gaining additional bandwidth savings at the client side. Our implementation utilizes the TCP Options field, supporting all TCP based applications such as audio, video streaming, P2P, etc.. We demonstrate that our solution achieves 30% redundancy elimination without significantly affecting the computational effort of the sender. This results in a 20% reduction of the overall cost to the cloud customer.

5. System Architecture

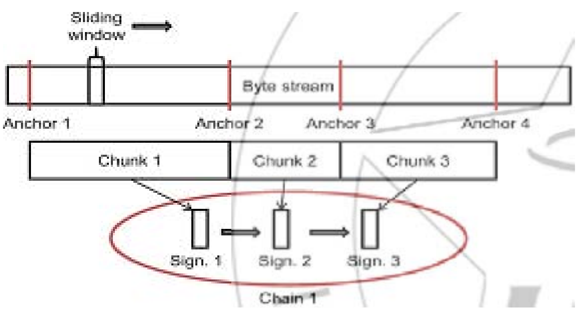

Figure 1: From stream to chain

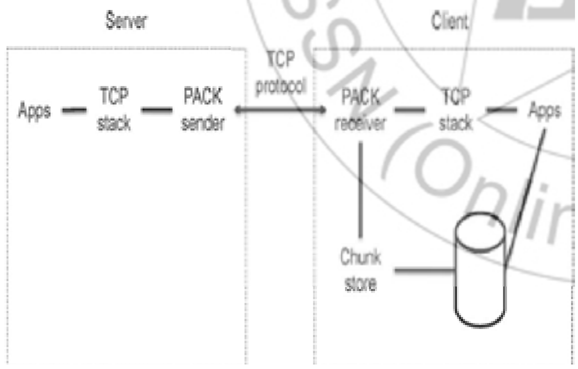

Figure 2: Overview of the PACK implementation.

6. Receiver Chunk Store

PACK uses new *chains* scheme in which data packets are linked to other small data packets according to their last received order. The PACK receiver maintains *a chunk store*, which is a large size cache of small chunks and their associated metadata. The chunk's metadata includes the chunks signature and a pointer which points to the next subsequent chunk in the. Here catching and indexing techniques are employed to efficiently maintain and retrieve the stored chunks signatures and the chains are formed by traversing the chunk pointers.

7. Receiver Algorithm

Upon the arrival of the new data, the receiver then computes the respective signature for each chunk and looks for a match in its local chunk store. If the chunk's signature is found then the receiver determines whether it is a part of a previously received chain, using the chunks' metadata. If yes, the receiver then sends a prediction to the sender for several next expected chain chunks. Upon a successful prediction, the sender then responds with a PRED-ACK confirmation message. Once the PRED-ACK message is received and processed, the receiver then copies the corresponding data from the chunk store to its TCP input buffers, placing it according to the corresponding sequence numbers. At this point, the receiver sends a normal TCP ACK with the next expected TCP sequence number. In case the prediction is false, the sender continues with normal operation, e.g., sending the raw data, without sending a PRED-ACK message.

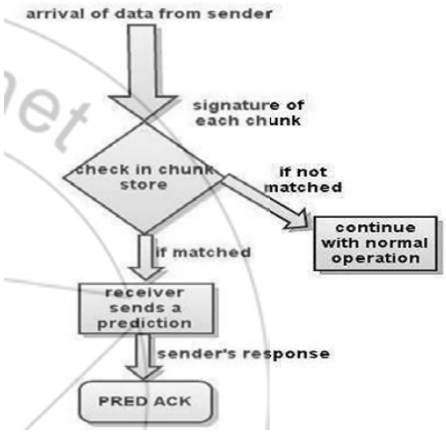

Figure 3: The receiver algorithm

8. Sender Algorithm

When the sender receives a PRED message from the receiver, it then tries to match the received predictions to its buffered (yet to be sent) data. For each prediction, the sender then determines the corresponding TCP sequence range and verifies the hint. If the hint match, the sender calculates the more computationally intensive SHA-1 signature for the predicted range of data and compares the result to the signature received in the PRED message. In case the hint does not match, a computationally expansive operation is saved. If the two SHA-1 signatures match, the sender assumes that the receiver's prediction is correct. In this case, it replaces the corresponding outgoing buffered data with a PRED-ACK message.

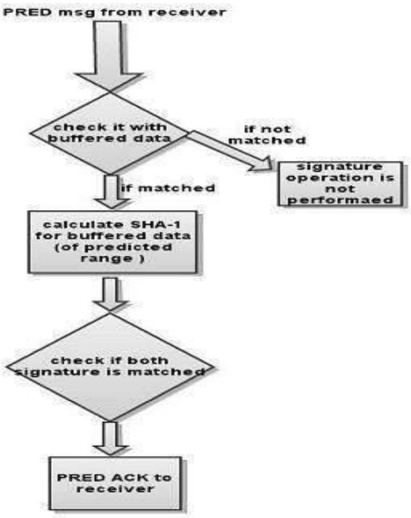

Figure 4: The sender algorithm

9. Wire Protocol

The TCP Options field is used to carry the PACK wire protocol. The PACK wire protocol operates under the assumption that the data is redundant. First, During the initial TCP handshake both the sides enable the PACK option by adding a *PACK permitted* to the TCP Options field. After that the redundant data is send by the sender in one or more TCP segments. The receiver identifies that a currently received chunk is identical to a previously received chunk in its chunk store. The receiver then, triggers a TCP ACK message and includes the prediction in the packet's Options field. Lastly, the sender sends a confirmation message (PRED-ACK) replacing the actual data.

10. Conclusion

Cloud computing triggers a high demand for TRE solutions as the amount of data exchanged between the cloud and its users is expected to increase. The cloud environment redefines the TRE system requirements, making the proprietary middle-box solutions inadequate. So, there is a rising need for a TRE solution that reduces the cloud's operational cost, for application latencies, user mobility and cloud elasticity. In this work, we have presented a Cost Reduction system based on prediction , cloud friendly end-to-end TRE based on novel speculative principles that reduces latency and the cloud operational cost. Here, PACK does not require the server to continuously maintain the clients' status, which enables cloud elasticity and user mobility while preserving the long term redundancy. An interesting extension of this work is the statistical study of chains of chunks which enables multiple possibilities in both the chunk order and the corresponding predictions. The system could also allow to make multiple predictions at a time and it is enough that one of them will be correct for successful 96% traffic elimination. A second promising direction can be the mode of operation optimization of the hybrid sender-receiver approach which is based on shared decisions derived from the receiver's power or the server's cost changes.

References

[1] E.Zohar, I.Cidon, O.Mokryn;"PACK: Prediction-Based Cloud Bandwidth and Cost Reduction System"; IEEE/ACM Transactions on Networking; 2013; 1063-6692 IEEE
[2] B.Aggarwal, A.Akella, A.Anand, A. Balachandran, P.Chitnis,G. Varghese C. Muthukrishnan, R. Ramjee , "EndRE: An end-system redundancy elimination service for enterprises," in Proc. NSDI,2010.
[3] C.Muthukrishnan, A. Anand, A. Akella, R. Ramjee. "Redundancy in Network Traffic: Finding and Implications" Proc. SIGMETRICS, 2009.
[4] U. Manber, "Finding similar files in a large file system," Proc. USENIX Winter Tech. Conf., 1994, pp. 1–10.
[5] A. Muthitacharoen, B. Chen, D. Mazières, "A low-bandwidth network file system," in Proc. SOSP, 2001, pp. 174–187.
[6] Ihm, K Park, V. Pai, "Wide area network acceleration for the developing world", in proc. USENIX ATC, 2010,pp 18-18
[7] Ashwini Ghorpade, S.R.Idate ," Survey of Traffic Redundancy Elimination systems for cloud computing environment to Reduce Cloud Bandwidth and Costs," in IJCTA, ISSN:2229-6093