
Traffic Redundancy Elimination System

S. Suresh Kumar & M.Divya

¹ Assistant Professor, Department of CSE, Balaji Institute of Technology & science, Warangal District, Telangana, India.

² M.Tech Student, Department of CSE, Balaji Institute of Technology & science, Warangal District, Telangana, India

Abstract:

In this paper, we present PACK (Predictive ACKs), a novel end-to-end traffic redundancy elimination (TRE) system, designed for cloud computing customers. Predictive Acknowledgments (PACK) is the latest cloud-based end-to-end Traffic Redundancy Elimination system (TRE). Cloud-based systems have to optimize the usage of resources such as bandwidth. The bandwidth usage is reduced by decreasing the cost of TRE computation and storage. ACK 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 server status does not have to be continuously maintained when using the PACK protocol. This feature makes PACK suitable for client mobility and server migration, which creates universal calculation environments responsible for maintaining cloud elasticity.

Keywords: Caching, cloud computing, network optimization, traffic redundancy elimination.

1. Introduction:

Cloud computing offers its customers an economical and convenient *pay-as-you-go* service model, known also as *usage-based pricing*. 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. The 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. 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. 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.

Disadvantages of Existing System:

1. Cloud providers cannot benefit from a technology whose goal is to reduce customer bandwidth bills, and thus are not likely to invest in one.
2. 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.

3. 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

4. Current end-to-end solutions also suffer from the requirement to maintain end-to-end synchronization that may result in degraded TRE efficiency.

3. Proposed System:

In this paper, 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. 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’ metadata 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. On the receiver side, we propose a new computationally lightweight chunking (fingerprinting) scheme termed *PACK chunking*. *PACK chunking* is a

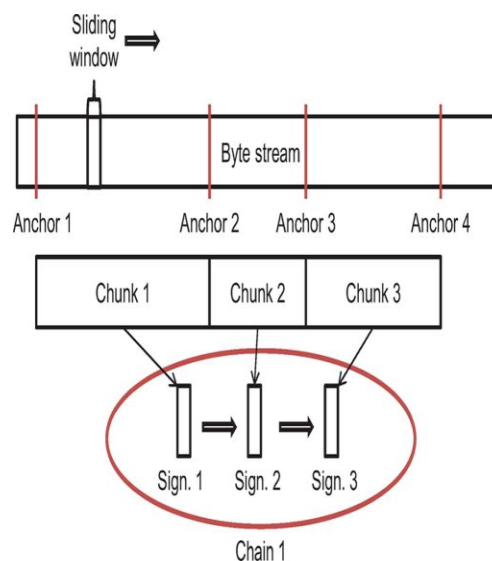
new alternative for Rabin fingerprinting traditionally used by RE applications.

Advantages of Proposed System:

1. Our approach can reach data processing speeds over 3 Gb/s, at least 20% faster than Rabin fingerprinting.
2. The receiver-based TRE solution addresses mobility problems common to quasi-mobile desktop/ laptops computational environments.
3. One of them is cloud elasticity due to which the servers are dynamically relocated around the federated cloud, thus causing clients to interact with multiple changing servers.
4. We implemented, tested, and performed realistic experiments with 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.
5. Our implementation utilizes the TCP Options field, supporting all TCP-based applications such as Web, video streaming, P2P, e-mail, etc.
6. We demonstrate that our solution achieves 30% redundancy elimination without significantly affecting the computational effort

of the sender, resulting in a 20% reduction of the overall cost to the cloud customer.

System Design:



- Receiver Chunk Store
- Receiver Algorithm
- Sender Algorithm
- Wire Protocol

Receiver Chunk Store

PACK uses a new *chains* scheme. Which chunks are linked to other chunks according to their last received order. The PACK receiver maintains a *chunk store*, which is a large size cache of chunks and their associated metadata.

Chunk's metadata includes the chunk's signature and a (single) pointer to the successive chunk in the last received stream containing this chunk. Caching and indexing techniques are employed to efficiently maintain and retrieve the stored chunks, their signatures, and the chains formed by traversing the chunk pointers.

When the new data are received and parsed to chunks, the receiver computes each chunk's signature using SHA-1. At this point, the chunk and its signature are added to the chunk store. In addition, the metadata of the previously received chunk in the same stream is updated to point to the current chunk. The unsynchronized nature of PRED allows the receiver to map each existing file in the local file system to a chain of chunks, saving in the chunk store only the metadata associated with the chunks.

Receiver Algorithm

Upon the arrival of new data, the receiver computes the respective signature for each chunk and looks for a match in its local chunk store. If the chunk's signature is found, the receiver determines whether it is a part of a formerly received chain, using the chunks' metadata. If affirmative, the receiver sends a prediction to the sender for several next expected chain chunks. Upon a successful

prediction, the sender responds with a PRED-ACK confirmation message. Once the PRED-ACK message is received and processed, the receiver 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, or one or more predicted chunks are already sent, the sender continues with normal operation, e.g., sending the raw data, without sending a PRED-ACK message.

Sender Algorithm

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

case, it replaces the corresponding outgoing buffered data with a PRED-ACK message.

Wire Protocol

The existing firewalls and minimizes overheads; we use the TCP Options field to carry the PACK wire protocol. It is clear that PACK can also be implemented above the TCP level while using similar message types and control fields. The PACK wire protocol operates under the assumption that the data is redundant. First, both sides enable the PACK option during the initial TCP handshake by adding a *PACK permitted* to the TCP Options field. Then, the sender sends the (redundant) data in one or more TCP segments, and the receiver identifies that a currently received chunk is identical to a chunk in its chunk store. The receiver, in turn, triggers a TCP ACK message and includes the prediction in the packet's Options field. Last, the sender sends a confirmation message (PRED-ACK) replacing the actual data.

4. Conclusion:

Cloud computing is expected to trigger high demand for TRE solutions as the amount of data exchanged between the cloud and its users is expected to dramatically increase. The cloud environment redefines the TRE system requirements, making proprietary middle-box

solutions inadequate. Consequently, there is a rising need for a TRE solution that reduces the cloud's operational cost while accounting for application latencies, user mobility, and cloud elasticity.

References:

- [1] E. Zohar, I. Cidon, and O. Mokryn, —The power of prediction: Cloud bandwidth and cost reduction,|| in Proc. SIGCOMM, 2011, pp. 86–97.
- [2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, —A view of cloud computing,|| Commun. ACM, vol. 53, no. 4, pp. 50–58, 2010.
- [3] U. Manber, —Finding similar files in a large file system,|| in Proc. USENIX Winter Tech. Conf., 1994, pp. 1–10.
- [4] N. T. Spring and D. Wetherall, —A protocol-independent technique for eliminating redundant network traffic,|| in Proc. SIGCOMM, 2000, vol. 30, pp. 87–95.
- [5] A. Muthitacharoen, B. Chen, and D. Mazières, —A low-bandwidth network file system,|| in Proc. SOSP, 2001, pp. 174–187.
- [6] E. Lev-Ran, I. Cidon, and I. Z. Ben-Shaul, —Method and apparatus for reducing network traffic over low bandwidth links,|| US Patent 7636767, Nov. 2009.

[7] S.MccanneandM. Demmer, —Content-based segmentation scheme for data compression in storage and transmission including hierarchical segment representation,|| US Patent 6828925, Dec. 2004.

[8] R. Williams, —Method for partitioning a block of data into sub blocks and for storing

[9] Blue Coat Systems, Sunnyvale, CA, USA, —MACH5,|| 1996 [Online].Available: <http://www.bluecoat.com/products/mach5>

[10] Expand Networks, Riverbed Technology, San Francisco, CA,USA, —Application acceleration and WAN optimization,||1998 [Online].Available:

<http://www.expand.com/technology/application-acceleration.aspx>

[11] F5, Seattle,WA, USA, —WAN optimization,|| 1996 [Online]. Available:<http://www.f5.com/solutions/acceleration/wanoptimization/>

[12] A. Flint, —The next workplace revolution,|| Nov. 2012 [Online].Available: <http://m.theatlanticcities.com/jobs-and->

economy/2012/11/nextworkplace-revolution/3904/



S.Suresh kumar is pursuing Ph.D in K.L University .He has 11+ years of teaching Experience. He is expert in software engineering and his research area includes cloud computing.



M. Divya Currently doing M.Tech in Computer Science & Engineering at BALAJI INSTITUTE OF TECHNOLOGICAL & SCIENCES-NARSAMPET, Warangal, India. Her research interests includes Cloud Computing, Network Security, Mobile Computing, etc,