International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

A Novel Approach for Cloud Bandwidth Reduction and Cost Reduction Using PACK

Rajalakshmi N.[1], Yogish H. K.[2]

[1, 2] Computer Science and Engineering Department, VTU Belgaum, EWIT, Bangalore, India

Abstract: ***The PACK (Predictive ACKs), a novel end-to-end traffic redundancy elimination (TRE) system, designed for cloud computing customers. Cloud-based TRE needs to apply a judicious use of cloud resources so that the bandwidth cost reduction combined with the additional cost of TRE computation and storage would be optimized. PACK's main advantages' its capability of offloading the cloud-server TRE effort to end clients, thus minimizing the processing costs induced by the TRE algorithm. Unlike previous solutions, PACK does not require the server to continuously maintain clients' status. The PACK is suitable for pervasive computation environments that combine client mobility and server migration to maintain cloud elasticity. 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.***

Keyword: cloud computing, traffic redundancy elimination, quality of service**,** caching, network optimization

1. Introduction

Cloud Cloud Computing, that is providing computer resources as a service, is a technology revolution offering flexible IT usage in a cost efficient and pay-per-use way. 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. 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 . 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 While proprietary middle-boxes are popular point solutions within enterprises, they are not as attractive in a cloud environment.. 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. 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. 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. The 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 and it require to maintain end-to-end synchronization that may degraded TRE efficiency. Here make use of an 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. Related Work

2.1 A low-bandwidth network file system

The LBFS is a network file system designed for low-bandwidth networks. LBFS exploits similarities between files or versions of the same file to save bandwidth. It avoids sending data over the network when the same data can already be found in the server's file system or the client's cache. Using this technique in conjunction with conventional compression and caching, LBFS consumes over an order of magnitude less bandwidth than traditional network file systems on common workloads.

2.2 SmartRE: An architecture for coordinated network-wide redundancy elimination

SmartRE, a practical and efficient architecture for network-wide RE. they show that SmartRE can enable more effective utilization of the available resources at network devices by reducing the wide-area footprint, and by improve end-to-end application performance. Therefore SmartRE can magnify the overall benefits of network-wide RE. SmartRE is naturally suited to handle heterogeneous resource constraints and traffic patterns and for incremental deployment.

2.3 Redundancy in network traffic: Findings and implications

The *protocol-independent redundancy elimination*, are used to improve network link performance by removing duplicate strings from within arbitrary network flows, has emerged as a powerful technique to improve the efficiency of network links in the face of repeated data. Many vendors offer such redundancy elimination middleboxes to improve the effective bandwidth of enterprise, data center and ISP links.

Licensed Under Creative Commons Attribution CC BY

2.4 EndRE: An end-system redundancy elimination service for enterprises

EndRE, is an alternative approach where redundancy elimination (RE) is provided as *an end system service*. Unlike middle boxes, such an approach benefits both end-to-end encrypted traffic as well as traffic on last-hop wireless links to mobile devices. EndRE uses a new fingerprinting scheme called SampleByte that is much faster than Rabin fingerprinting while delivering similar compression gains. Unlike Rabin fingerprinting, SampleByte can also adapt its CPU usage depending on server load. end-to-end latencies by up to 30%, and translates bandwidth savings into equivalent energy savings on mobile smart phones.

3. Proposed System

PACK is a new *chains* scheme here the 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 PACK 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**.** 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.

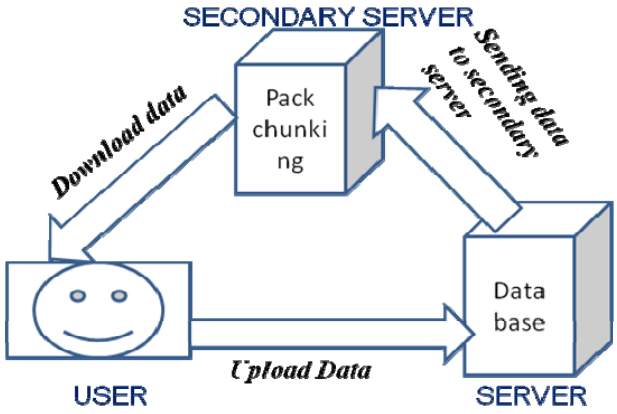

Figure 1: System Architecture

3.1 Module description

a) User module
User module is used to upload the data and download the data from server and secondary server.

b) Server module
Server module receives the data or file from the user and partition the data based on previous history and store in different chunk storage system.

c) Secondary server module
This module downloads the partition data or file from the server and eliminates the redundant file by using the prediction acknowledgement and then merge the partition file.

3.2 Implementation details

a) User Module

1. User first establish connection with cloud server
2. User login with user name and password
3. User uploads the data to the cloud server
4. User then connect to secondary server and download the data user module

b) Cloud server module

1. Server first establish connection between chunk storage
2. Server accepts the connection request of the user
3. Server obtains the uploaded data and partition the uploaded data into equal number of chunks and store in chunk storage.

c) Secondary server module

1. The secondary server module first establishes the connection with cloud server database.
2. secondary server downloads the data or file from the server database based on user request.
3. secondary server uses the PACK (predictive acknowledgement) to eliminate the redundant data transmission.
4. secondary server merges the data according to sequence number then complete file is downloaded successfully.

d) Advantages

- PACK's main advantage is its capability of offloading the cloud-server TRE effort to end clients, thus minimizing the processing costs induced by the TRE algorithm
- Reduce latency and cloud operational cost.
- Doesn't require the server to continuously maintain client's status, thus enabling cloud elasticity and user mobility.

4. Conclusion

The proposed PACK is a receiver-based, cloud-friendly, end-to-end TRE that is based on novel speculative principles that reduce latency and cloud operational cost. PACK does not require the server to continuously maintain clients' status, thus enabling cloud elasticity and user mobility while preserving long-term redundancy. Moreover, PACK is capable of eliminating redundancy based on content arriving to the client from multiple servers without applying a three-way handshake.

Licensed Under Creative Commons Attribution CC BY

5. Acknowledgment

We express our heartfelt sincere gratitude to HOD of Computer Science and Engineering, East West Institute of Technology for his valuable suggestions and support. Special thanks to coordinator and guide for their valuable support and guidance.

References

[1] Muthitacharoen, B. Chen, and D. Mazières, "A low-bandwidth network file system," in *Proc. SOSP*, 2001, pp. 174–187.
[2] A. Gupta, A. Akella, S. Seshan, S. Shenker, and J. Wang, "Understanding and exploiting network traffic redundancy," UW-Madison, Madison, WI, USA, Tech. Rep. 1592, Apr. 2007.
[3] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker, "Packet caches on routers: The implications of universal redundant traffic elimination," in *Proc. SIGCOMM*, 2008, pp. 219–230.
[4] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, "Redundancy in network traffic: Findings and implications," in *Proc. SIGMETRICS*, 2009, pp. 37–48.
[5] B. Aggarwal, A. Akella, A. Anand, A. Balachandran, P. Chitnis, C.Muthukrishnan, R. Ramjee, and G. Varghese, "EndRE: An end-system redundancy elimination service for enterprises," in *Proc. NSDI*, 2010, pp. 28–28.
[6] A. Anand, V. Sekar, and A. Akella, "SmartRE: An architecture for coordinated network-wide redundancy elimination," in *Proc. SIGCOMM*, 2009, vol. 39, pp. 87–98.

Licensed Under Creative Commons Attribution CC BY