ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

Periodically Predicting Client's Bandwidth & Cost Acknowledgements Sends to Cloud Server to Optimize Resource Usage

[1]Prakash E J, [2] A.Nageswara Rao

[1]M.Tech Student, Department of Computer Science and Engineering, SV College of Engineering, Tirupati, Chittoor, Andhra Pradesh, India

[2] Professor & Head, Department of Computer Science Engineering, SV College of Engineering Tirupati, Chittoor, Andhra Pradesh, India

ABSTRACT: Cloud computing provides users to store their data remotely and enjoy the on demand high quality services and offers it as usage based pricing. Here, bandwidth reduction is an important issue in this paper. Cloud service providers (CSP) are trying to reduce this bandwidth and also applying the judicious use of cloud resources by deploying a TRE system to end-to-end clients. Generally cloud server maintains the client mobility and server migration to cloud elasticity along with this it also maintaining the continuous client's status of data transferring, traffic reduction, it is again a burden to server, So in order to reduce the overhead to server in this paper TRE system is receiver based, by maintaining a chunk store and allowing the client to use newly received chunks to identify previously received chunk chains which in turn sends a predictor message for the subsequent chunks. When redundancy is detected the sender then sends only ACK to the prediction instead of sending data.

KEYWORDS: caching, cloud computing, network optimization, traffic redundancy elimination.

I. INTRODUCTION

The recent surge in cloud computing arises from its ability to provide software, infrastructure, and platform services without requiring large investments or expenses to manage and operate them. Cloud computing characteristics include a ubiquitous (network-based) access channel; resource pooling; multitenancy; automatic and elastic provisioning and release of computing capabilities; and metering of resource Usage (typically on a pay-per-use basis). 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.

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 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. On the other hand, cloud-side elasticity motivates work distribution

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

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.

We 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. Moreover, if an end-to-end solution is employed, its additional computational and storage costs at the cloud side should be weighed against its bandwidth saving gains. 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, we 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. Our experiments 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, 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. 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.

II. RELATED WORK

Several TRE techniques are there, independent TRE, packet-level TRE. These have combined the sender-based TRE ideas of with the algorithmic and implementation approach of along with protocol specific optimizations for middle-boxes solutions. In particular, describes how to get away with three-way handshake between the sender and the receiver if a full state synchronization is maintained. Some papers present redundancy-aware routing algorithm. These papers assume that the routers are equipped with data caches, and that they search those routes that make a better use of the cached data.

In existing system they formally describe the design of SmartRE, architecture for redundancy elimination that draws on the principles of spatially decoupling encoding and decoding responsibilities, and coordinating the actions of RE devices for maximum efficiency. Our description focuses on SmartRE as applied to an ISP network. SmartRE synthesizes two ideas: packet caches for redundancy elimination and cSamp. SmartRE leverages ideas from cSamp to split caching (and decoding) responsibilities across multiple router hops in a network. It specifies the caching responsibility of each RE device in terms of a hash-range per path per device. Each device is responsible for caching Spackets such that the hash of the packet header falls in its assigned ranges. By using the same hash function across the network and assigning non overlapping hash ranges across devices on the same path, SmartRE leverages the memory resources efficiently without requiring expensive cache coordination protocols.

Here in this design of three key elements ingress nodes, interior nodes, and a central configuration module. Ingress and interior nodes maintain caches storing a subset of packets they observe.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

1. Ingress nodes encode packets. They search for redundant content in incoming packets and encode them with respect to previously seen packets using some mechanism. In this sense, the role of an ingress node is identical in the naive hop-by hop approach and SmartRE.
2. First, interior elements need not store all packets in their packet cache – they only store a subset as specified by a caching manifest produced by the configuration module. Second, they have no encoding responsibilities. Interior nodes only decode packets, i.e., expand encoded regions specified by the ingresses using packets in their local packet cache.
3. The configuration module computes the caching manifests to optimize the ISP objective(s), while operating within the memory and packet processing constraints of network elements. Similar to other proposals for centralized network management we assume that this module will be at the network operations center (NOC), and has access to the network's traffic matrix, routing policies, and the resource configurations of the network elements.

Algorithm1: PROCESSPACKETINGRESS (pkt, ingress)

```
1 egress ← FINDEGRESS (pkt)
2 pathid ← GETPATHID (ingress, egress)
3 candidates ← GETCANDIDATES (pathid)
4 encodedpkt ← ENCODE (pkt, candidates)
5 coveredrange ← GETCOVEREDRANGE (pathid)
6 h ← HASH (pkt.header)
7 if (h ∈ coveredrange) then ADDPKTTOSTORE (pkt, pathid, h)
8 FORWARD (encodedpkt)
```

Pseudo code for ingress node.

Algorithm 2: PROCESSPACKETINTERIOR (encodedpkt, r)

```
1 mymatches ← PROCESSHIM (encodedpkt .shim)
2 decodedpkt ← DECODE (encodedpkt, mymatches)
3 pathid ← GETPATHID (encodedpkt)
4 myrange ← GETRANGE (pathid, r)
5 h ← HASH (pkt .header)
6 if (h ∈ myrange) then ADDPKTTOSTORE (decodedpkt, pathid, h)
7 FORWARD (decodedpkt)
```

Pseudo code for ingress node.

III. PROBLEM DEFINITION

In this paper we describe the basic receiver driven operation of the PACK protocol. 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.

Receiver chunk store:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

The system uses a new chains scheme, described in Fig. 1, in 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 network redundant traffic, not only consuming network bandwidth, but also reducing the efficiency of the Internet, exists abundantly in the course of exchanging Internet data. In the context, protocol-independent redundancy elimination, which can remove 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 and has been widely used. Redundancy elimination middle-boxes are being widely deployed to improve the effective bandwidth of network access links of enterprises and data centers alike, and for improving link loads in small ISP networks.

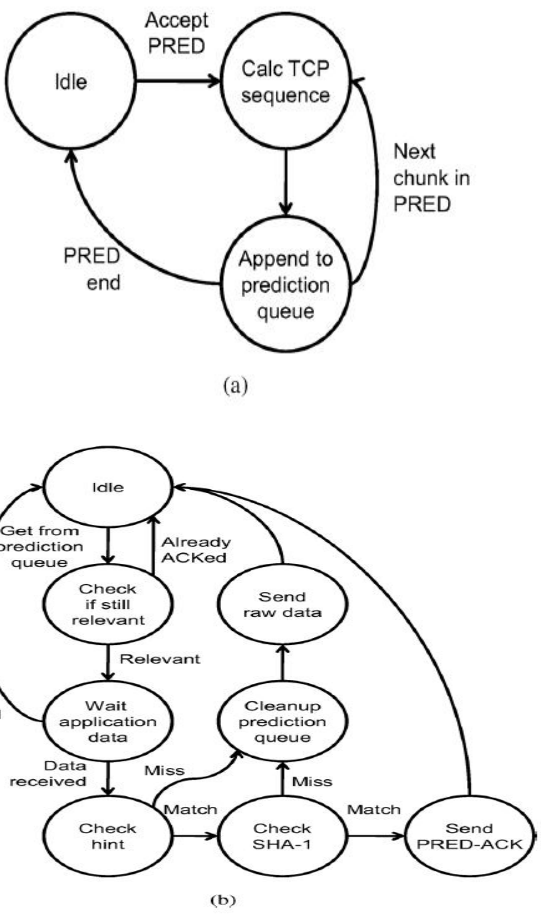


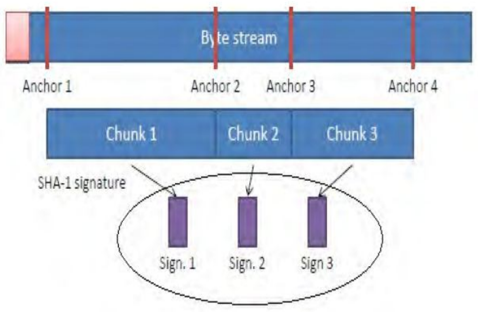
Fig 2: sender algorithms. (a)Filling the prediction queue (b) processing the prediction queue and sending PRED-ACK or raw data.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014



The utilization of a small chunk size presents better redundancy elimination when data modifications are fine-grained, such as sporadic changes in an HTML page. On the other hand, the use of smaller chunks increases the storage index size, memory usage, and magnetic disk seeks. It also increases the transmission overhead of the virtual data exchanged between the client and the server.

IV. ALGORITHMS

When a new data arrived, then 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. The prediction carries a starting point in the byte stream (i.e., offset) and the identity of several subsequent chunks (PRED command).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.

Algorithm. 3: Receiver Segment Processing
```
1. if segment carries payload data then
2. Calculate chunk
3. if reached chunk boundary then
4. activate predAttempt ()
5. end if
6. else if PRED-ACK segment then
7. ProcessPredAck ()
8. activate predAttempt ()
9. end if
```

Algorithm. 4: predAttempt ()
```
1. if received chunk matches one in chunk store then
2. if foundChain (chunk) then
3. Prepare PREDs
4. send single TCP ACK with PREDs according to
   Options free space
5. exit
6. end if
7. else
8. store chunk
9. link chunk to current chain
```

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

```
10.end if
11. send TCP ACK only
```

```
Algorithm. 5: processPredAck ()
1. for all offset PRED-ACK do
2. read data from chunk store
3. put data in TCP input buffer
4. End for
```

Fig 3, Shows the redundancy in each month, according to the e-mail message's issue date. The total measured traffic redundancy was 31.6%, which is roughly 350 MB. We found this redundancy to arise from large attachments that are sent by multiple sources, e-mail correspondence with similar documents in development process, and replies with large quoted text. This result is a conservative estimate of the amount of redundancy in cloud e-mail traffic because in practice some messages are read and downloaded multiple times. For example, a Gmail user that reads the same attachment for 10 times, directly from the Web browser, generates 90% redundant traffic.

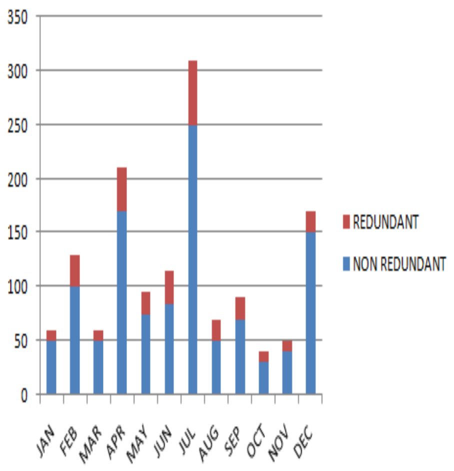


FIG: 3 Traffic volume and detected redundancy E-mail one year gmail account by month.

Furthermore, this evaluations show that in videos and large files with a small amount of changes, redundant chunks are likely to reside in very long chains that are efficiently handled by a receiver-based TRE.

V. Conclusion

In cloud computing mainly traffic redundancy elimination (TRE) is the main issue to concern as the amount of Data exchanged between the cloud and the users is increasing dramatically using middle-boxes in cloud environment are inadequate. So, the increasing need for TRE solution that reduces the operational cost, user mobility and cloud elasticity

In this paper we have presented a novel receiver based cloud friendly end-to-end TR that is based on new principals that reduce latency and cloud operational cost.

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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

[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.Mccanne andM. 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 subblocks and for storing and communicating such subblocks," US Patent 5990810, Nov. 1999.

BIOGRAPHY

Prakash E J received B.Tech Degree from Chadalawada Ramanamma Engineering College, Tirupati. He is currently pursuing M.Tech Degree in Computer Science Engineering specialization in SV College of Engineering, Tirupati, Andhra Pradesh, India.

A.Nageswara Rao received B.Tech Degree from CBIT, Osmaniya university, Hyderabad. M.Tech Degree in Computer Science at Central university, Hyderabad. He is currently working towards PhD in Data Mining specialization at Rayalaseema University, Kurnool, Andhra Pradesh, India, and working as professor & Head in the Dept. Of CSE, SV College of Engineering, Tirupati, Andhra Pradesh, India. He presented many research papers in National & International Conferences.