A Review On A Load Balancing Technique For Cost Reduction In Cloud Data Transmission

Neethu Mariam Joseph
Assistant Professor, Dept. of Information Technology
Karpagam College of Engineering
Coimbatore, India

Abstract—The cloud computing has become ubiquitous, casting its shadow over almost every facet of business processes within every industry. Its triumph is due to customers' capability to use services on demand with a pay-as-you go rating model, which has proved appropriate in many respects. Traffic redundancy elimination (TRE) is used to eliminate the redundant content transmission to decrease bandwidth costs. Redundant data chunks are sensed and detached. However for server specific TRE approach it is problematic to handle the traffic efficiently. Source based TRE delivers substantial loads to Server. This gives a TRE solution which does not need a server to maintain client position. The system is known as PACK "Prognostic-ACKnowledgements" which is receiver motivated. In this elucidation each receiver perceives the received stream and attempts to match its chunks with a formerly received chunk chain and sending to the server for forecasting the future data. On getting a match the receiver inquires the source if the subsequent chunks will be similar as the ones in its stock by sending the uniqueness of several chunks. This is known as a PACK command. If this is right the source sends PRED command-ACK and the receiver directs the chunks to its own TCP input buffers. If the forecast is not correct the source continues to send as usual.

Keywords-cloud computing; cost reduction; data transmission

I. INTRODUCTION

Cloud computing is the distribution of computing services over the Internet. Cloud services allow individuals and industries to use software and hardware that are achieved by third parties at remote locations. Examples of cloud services contain online file storing, social networking sites, webmail, and online business applications. The cloud computing model lets to access of the information and on the computer resources from anywhere that a network linking is available. Cloud computing offers a shared pool of resources, which includes the data storage space, networks, computer processing power, and dedicated corporate and user applications.

The features of cloud computing include on-demand self service, pooling of resources, wide network access, rapid elasticity and other measured service. On-demand self service means that customers can be able to request and manage their own computing resources [1]. The broad network access permits the services to be obtainable over the Internet or on the private networks. Pooled resource means that customers draw from a pool of computing resources, typically in remote data centers. The cloud services can be scaled larger or smaller; and use of a service is measured and clients are billed accordingly[2].

II. OBJECTIVE OF THE PROJECT

With the accumulative burdens of modern business, corporations and customers require fast outcomes, and should not have to pause on applications, websites or on their services. Load balancing is a solution that accomplishes resources and guides them to the suitable location. Put technically, load balancing ensures an even division of processing power amongst the network links, computer nodes, and on devices. Creating an effective load balancing architecture promotes exceptional application scalability and redundancy, by removing potential interruptions, downtime and confirming to a positive experience for both users and customers.

III. EXISTING SYSTEM

Network traffic displays large extent of redundancy when diverse users on the Internet access the similar content. Thus, TRE can be used to eliminate the progress of redundant data content and, therefore, to significantly reduce the network cost. In the most of the common TRE results, both the source and the receiver examine and compare signatures of data chunks, parsed according to the content prior to its transmission. When these redundant chunks are noticed, the source replaces the transmission of each redundant chunk with its solid signature value. Commercial TRE solutions are popular at enterprise networks, and involve the positioning of two or more exclusive protocols, state synchronized middle-boxes at both the intranet entry points of data hubs and branch offices, for eliminating repetitive traffic between them [3].

While exclusive middle-boxes are general point solutions within enterprises, they are not as attractive in a cloud environment. Current end-to-end TRE solutions are source-based. In the case where the cloud server is the source, these solutions needs that the server continuously maintains clients' status[4]. The popularity of rich media that consume high bandwidth stimulates content distribution network (CDN) solutions, in which the service point for static and mobile users may change animatedly according to the relative service point locations and loads[5].

A. EndRE : An end- system redundancy elimination service for enterprises methodology

The methodology this system uses is Redundancy Elimination (EndRE) and Fingerprinting scheme (Sample Byte) [6].

The advantages of this methodology include:

- Faster than the commonly used Rabin fingerprint.
- EndRE, an alternate approach where redundancy elimination (RE) is provided as an end system service.
- Benefits both end-to-end encrypted traffic as well as traffic on last-hop wireless links to mobile devices.
- EndRE delivers 26% bandwidth savings on average, processes payloads at speeds of 1.5-4Gbps, reduces end-to-end latencies by up to thirty percentage, and it translates bandwidth savings into equivalent energy savings on mobile Smartphone.

The disadvantages of this methodology include:

- WAN optimizers are being deployed in order to eliminate redundancy.
- Increase network costs and latency due to the resultant increase in WAN traffic.
- Rabin fingerprint-very slow.

B. A low bandwidth network file system

The methodology this system uses is LBFS, a network file system designed for low-bandwidth networks and hashing to exploit inter-file similarities and save bandwidth [7].

The advantages of this methodology include:

- LBFS consumes over an order of magnitude less bandwidth than traditional network file systems on common workloads.
- Provide robustness despite client failures.
- LBFS avoids transmitting redundant data to a remote machine
- Performs much better than the competition.
- Efficiently divide files into "chunks" and only transfer chunks the remote machine does not already have.

The disadvantages of this methodology include:

- Network file systems cannot be utilized over slower WANs due to longer delay, less available bandwidth
- Ad hoc solutions:
    - Copy file to local machine, edit, then upload. Risks update conflicts.
    - Use remote login. Potentially large interactive delays.

C. RACS : A case for cloud storage diversity

The methodology this system uses is called as redundant array of cloud storage (RACS) [8].

The advantages of this methodology include:

- A cloud storage proxy that transparently stripes data across multiple cloud storage providers.
- RACS enables cloud storage customers to explore trade-offs between overhead and mobility, and to better achieve the new progresses in the fast-paced cloud storage marketplace.

The disadvantages of this methodology include:

- The increasing popularity of cloud storage is leading organizations to consider moving data out of their own data hubs and into the cloud
- It develops very expensive for organizations to switch storage providers.
- Thus, a client progressing from one provider to another fee for bandwidth twice.

D. Packet Caches on Routers: The Implications of Universal Redundant Traffic Elimination

The methodology this system uses is redundancy-aware routing and inter-domain routing [9].

The advantages of this methodology include:

- Packet-level redundant data content elimination as a common primitive on all Internet routers.
- Universal deployment would immediately reduce link loads everywhere.
- ISPs much better control over link loads, a great degree of flexibility in meeting traffic engineering objectives, and greater ability to offer consistent performance under sudden traffic variations.

The disadvantages of this methodology include:

- Existing RE approaches apply only to point deployments. e.g. at stub network access links.
- They only benefit the system to which they are directly connected.

E. SMARTRE: Architecture for Coordinated Network-Wide Redundancy Elimination

The methodology this system uses is SmartRE [10].

The advantages of this methodology include:

- SmartRE is naturally suited to handle heterogeneous resource restraints and traffic patterns and for incremental deployment.
- SmartRE, a practical and active architecture for network-wide RE.
- SmartRE can enable more effective utilization of the available resources at network devices, and thus can magnify the complete aids of network-wide RE.

The disadvantages of this methodology include:

- Application-independent Redundancy Elimination (RE), or classifying and removing repetitive data content from network transmissions, has been utilized with great success for refining network performance on the enterprise access links.
- A network-wide RE service aids ISPs by lowering the link loads and aggregating the effective network capacity to well house the increasing number of bandwidth-intensive applications.
- Additional, a network wide RE service democratizes the benefits of RE to all end-to-end traffic and advances application performance by rapidly accumulating throughput and dropping the latencies. While the idea of a network-wide RE service is pleasing, realizing it in practice is challenging.

F. WINNOWING: Local Algorithms for Document Fingerprinting Methodology

The methodology this system uses is local document fingerprinting algorithms and winnowing [11].

The advantages of this methodology include:

- An efficient local fingerprinting algorithm and show that winnowing performance is within 33% of the lower bound.
- Capture an essential property of any fingerprinting technique guaranteed to detect copies.

The disadvantages of this methodology include:

- Less accuracy with accurately identifying copying, including small partial copies, within large sets of documents.

IV. PROPOSED SYSTEM

The new traffic redundancy elimination approach also called PACK (Prognostic ACKs) or a novel end-to-end traffic redundancy elimination (TRE) system, which detects redundancy at the client side and there is no need of server to continuously. The basic unit of data content here is called as the chunk. The receiver develops a chunk store which operates by the LRU (Line Replacement Unit) principle.

Chunks are recognized by “hints” which are easily computed functions which identify data content with a fast algorithm with low false positives. If this hints the match, then SHA-1 signatures are checked and if matched then data about successive chunks (referred to as a “chain”) can be got back from pointers in chunk data. So, a chunk is the data with a signature to uniquely recognize the data contents and a link to the next chunk (if received). On receiving a match the receiver requests the source if the next chunks will be the same as the ones in its store by transferring the identity of several chunks. This is known as a PACK command (predictive). If this is correct the source sends PACK command-ACK and the receiver leads the chunks to its own TCP input buffers. If the prediction is not correct the source continues to lead as normal. The TCP Options field is utilized to carry the PACK wired protocol. In order to get optimal resource consumption and reduction in computing time load balancing is implemented in this system. This also helps to increase the capacity of a server farm beyond that of a single server.

Advantages of the system

PACK saves cloud's computational power in the absence of redundancy.

Reduces processing costs induced by TRE - Signs with SHA-1 in the existence of redundancy.

Suitable for cloud server elasticity and client mobility – Does not involve the server to continuously maintain clients' status.

A cloud cost reduction is attained at a practical client effort although gaining added savings of bandwidth at the client side.

Increasing the cloud performance and decrease the transmission delay using load balancing technique.

One of them is cloud elasticity due to which the servers are with dynamism relocated around the associated cloud, thus causing clients to interact with multiple changing servers.

A. PACK Alogorithm

Receiver algorithm

Arrival of new data receiver computes the respective signature of each chunk. If match found in chunk stock then receiver decides whether it is a part of a formerly received chain using chunk's metadata. If yes, then receiver sends a prediction of starting point in the byte stream. The offset and the identity of several subsequent chunks (PACK command). Upon a successful prediction, source responds PACK command -ACK. Then receiver copies the consistent data content from the chunk stock to its TCP buffer, according to the consistent sequence number. Then source responds by normal TCP ACK with the next probable TCP sequence number. If estimation is false, Source continues with normal operation e.g. sending the raw data, without sending a PACK command-ACK message.

Source Algorithm

Upon receiving PACK command message from the receiver, which then tries to match it with its buffered data. Then, upon a hint match the source calculates SHA-1 signature for the predicted range. It then compares the result to the signature received in the PACK command message. If both signature matched, source can safely assume that the receiver's prediction is correct then PACK command-ACK message is sent. If hint not matched then source sends requested data to receiver.

PACK Chunking

XOR based rolling hash function, for fast TRE chunking Anchors are noticed by mask the at provides on average 8KB chunk and 48bytes in the sliding window PACK chunking is quicker due to one less XOR operation per byte.

B. Overview of the PACK Implementation

The figure 1 represents the overview of how the PACK implementation process takes place and also about the chunk store and figure 2 shows the data content flow.

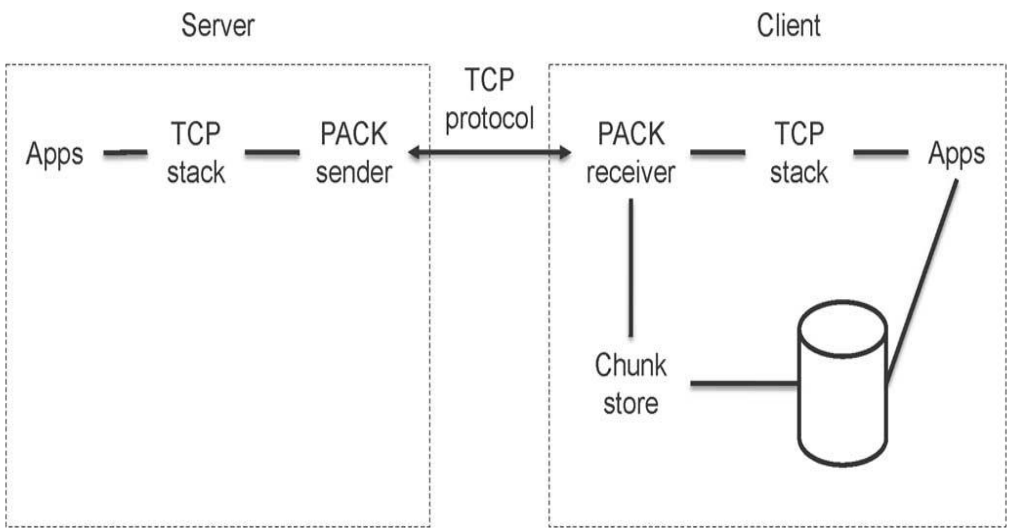

Figure1 Pack implementation representation

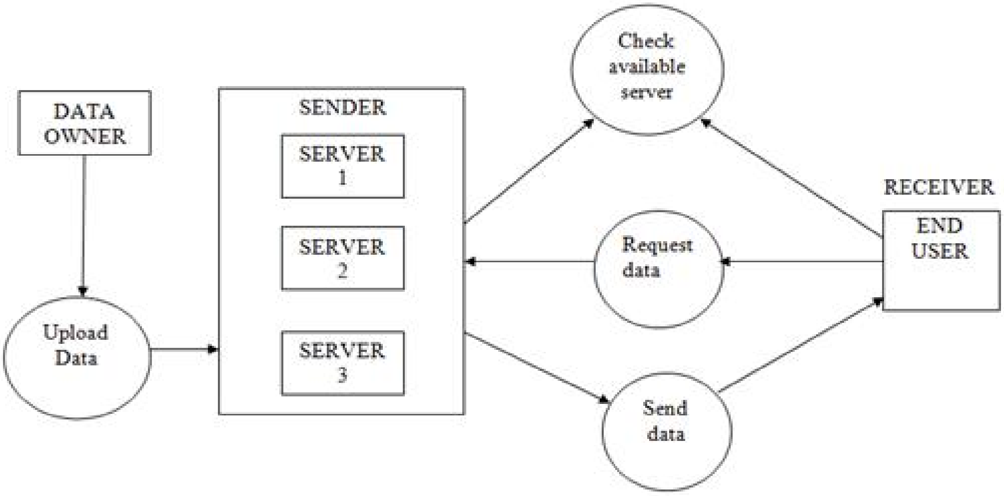

Figure 2 Level 0 Dataflow

C. Implementation Description

1) Module 1- Data owner:

Data owner can export data to the cloud. In this module data owner can become the cloud customer by creating an account. Data owner can Upload the file into cloud server and then get the details of the cloud cost based on storage and data transfer cost (i.e. bandwidth).

2) Module 2- Cloud user:

Cloud users are end users who can download or request data. In this module user has to create an account and then login. And view the files list uploaded by data owner. User can request files to server.

3) Module 3- Chunks:

Source sends the requested data to the receiver. Receiver maintains chunk store. Once the new data are received, parsed to chunks then the receiver computes signature using SHA-1. Chunk store has the metadata of each chunk includes the chunk's signature and a pointer to the subsequent chunk in the last received data stream.

4) Module 4 - Prediction mechanism:

If the source sends the (redundant) data, then the receiver computes the respective signature for every chunk and looks for a match with previously received chunks in its local chunk store. If it matches then the PACK command (includes the signature and hint of predicted data) for several next expected chain chunks is sent to source.

5) Module 5- Cloud Server:

Cloud server will allocate some cloud bandwidth to the user after that data owner is able to upload the file into the server and server sends details of usage for every transaction.

6) Module 6- Prognostic –ACK:

Server receives the prediction command then it tries to match the received predictions to data yet to send. If the hint matches then the server computes the SHA-1 signature for the data and compares with the Predicted message sent by the receiver. If it matches server sends Predictive Acknowledge to the receiver. Otherwise it will send the data. Once the PACK command-ACK message is received, the receiver copies the corresponding data from the chunk store.

7) Module 7- Load balancing:

This module helps to manage overload by calculating the response time of multiple servers. Then user can request the data to the server which has minimum response time. This will maximize the workload performance and help prevent overload.

V. CONCLUSION AND FUTURE WORK

Cloud computing probably triggers a high demand for TRE solutions as the amount of data replaced between the cloud and its users is likely to dramatically rise. The cloud environment redefines the TRE system supplies, making proprietary middle-box resolutions inadequate. Consequently, there is a growing need for a TRE solution that decreases the cloud's working cost while accounting for application latencies, user mobility, and cloud elasticity.

Thus PACK is a cloud friendly receiver-based, end-to-end TRE that is based on novel abstract principles that reduce latency and cloud operational cost. PACK does not want the server to continuously keep clients' status, thus enabling cloud elasticity and user movement while conserving long-term redundancy. Moreover, PACK is capable of eliminating redundancy based on content arrival to the client from several servers without having a three-way handshake.

The evaluation use a wide collection of data content types that displays the PACK which meets the expected design goals and has clear advantages over source-based TRE, especially when the cloud computation cost and buffering necessities are important. Moreover, PACK imposes additional effort on the source only when redundancy is exploited, thus reducing the cloud overall cost. Two interesting future extensions can deliver added aids to the PACK concept. First, the implementation maintains chains by keeping for any chunk only the latter detected following chunk in an LRU style.

An interesting extension to this work is the statistical learning of chains of chunks that would allow multiple options in both the chunk order and the corresponding predictions. The system may also permit making extra predictions at a time, and it is enough that one of them will be correct for effective traffic elimination. A second hopeful direction is the mode of operation optimization of the hybrid source–receiver method based on collective decisions resulting from receiver's power or server's cost changes.

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] S.Mccanne and M. Demmer, "Content-based segmentation scheme for data compression in storage and transmission including hierarchical segment representation", US Patent 6828925, Dec. 2004.
[6] 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.
[7] A. Muthitacharoen, B. Chen, and D. Mazières, 'A low-bandwidth network file system', in Proc. SOSP, 2001, pp. 174–187.
[8] H. Abu-Libdeh, L. Princehouse, and H.Weatherspoon, "RACS: A case for cloud storage diversity", in Proc. SOCC, 2010, pp. 229–240.
[9] 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.
[10] 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.
[11] S. Schleimer, D. S. Wilkerson, and A. Aiken, "Winnowing: Local algorithms for document fingerprinting", in Proc. SIGMOD, 2003, pp.76–85.