International Journal of Advanced Technology in Engineering and Science www.ijates.com
Volume No.03, Issue No. 06, June 2015 ISSN (online): 2348 – 7550

A TRIVIAL SCHEME FOR ELIMINATING REDUNDANT RESOURCES ON CLOUD

P. Hari Priya[1], P. Lavanya[2]

[1]Assistant Professor, Dept. of CSE Student, [2]Assistant Professor, Dept. of CSE,
S. V. Engineering College for Women, Tirupati, (India)

ABSTRACT

An idea on eliminating the duplicate resources on cloud by checking small portion of the resources by comparing with the already downloaded stuff, such that a lot of burden on the Client would get relieved like Price, Time, and Effort etc… The main intention is to offload the server in checking for duplicates such that the client itself can check for duplicate resources using predictive acknowledgement. The concept also enhances the idea by providing cloud Client and Server Migration, Redundant Resource Elimination. It's an absolute new intention that earlier would rely on the solution like Name matching, comparing old packets with the new packages arrived just. The previous solution is based on TCP and extension has been made to work on all Platforms and Protocols.

Keywords: Bandwidth Optimization, Cloud Elasticity, Cloud Resources Optimization, Redundant Resources.

I. INTRODUCTION

As Market grows we all knew that Cloud Computing is a very vast and emerging means of providing services to the client in its various forms such as platform, infrastructure, and software. In addition to this now they added some extensions like Security as a service, Database as a Service, etc… As each and every firm or organization offers its services through cloud in the present and very extensively in the nearby future all the Company's Services were available as a part of Cloud. So we got Cloud services as PAY PER USE service. Clients or customer pay only to their usage regarding the Bandwidth, CPU power, Servers using , Processing Power, VMs Consumption, Memory Consumption etc… Data transfer costs also include heavy impact on the Cloud Services if the users download the resources which were already downloaded that might be with a different name or same name. If the redundant services are not removed from the Server very clearly they would increase the network traffic which may cause severe issues. Earlier the complete load is on the server which has to check whether any duplicate resource is existing or not. Actually this is a new idea when the server is not in use but the Server can't be expected to be free all the time. In fact Server is usually busy in serving the clients so checking or redundant resources on the server side would not worked for a longer time. Next Idea is to provide these checking both at server and client side. This also became vague over a period. Finally the server is off loaded from this checking where that job is taken care by the Client itself such that whenever it found such redundant packets or data it will send an acknowledgement to the server (predictive acknowledgement).

II. RELATED WORK

Many Techniques have been developed over the years for redundant packet elimination but initially that is done on purely server side only. Later it was extended to Server and Client, finally that was implemented on client side. This paper avoids all the third party middle-box techniques for finding the duplicate packets. Hash code will be generated for each and every packet, stored on the table and for each new packet this hash code will be evaluated and compared with the hash codes which are already stored. If the code is found it indicates the packet has been duplicated and a prediction is sent to the server for this duplicate packet. At last the Server will scrutinize the packet and sends an acknowledgement to the Client.

III. IDENTIFICATION OF PROBLEM

Generally in cloud based services checking for the redundant copies of resource is done on the server side which is a expensive task in terms of cost and effort. This can be shared both on server and client side, this is a little bit simplified version when compared to checking the whole thing on the server side. This complete mechanism should be shifted to on the client side. The previous solution doesn't address the problems like Cloud Elasticity, Resource Optimization. The Problem is the checking is done based on the name but on the content that whether it has been duplicated.

A Heavyweight model for eliminating the duplicates on the cloud using Server Side checking, Server and client side checking, Rabin Fingerprinting methods seems not applicable to a greater extent to the real time problems. Traffic Redundancy will be increased a lot and can't be controlled with these ideas. These will increase the Cost, Traffic, Bandwidth Consumptions, end users effort and Time. The identified problems are very fatal which needs immediate concern for handling and setting up right.

IV. SYSTEM ARCHITECTURE

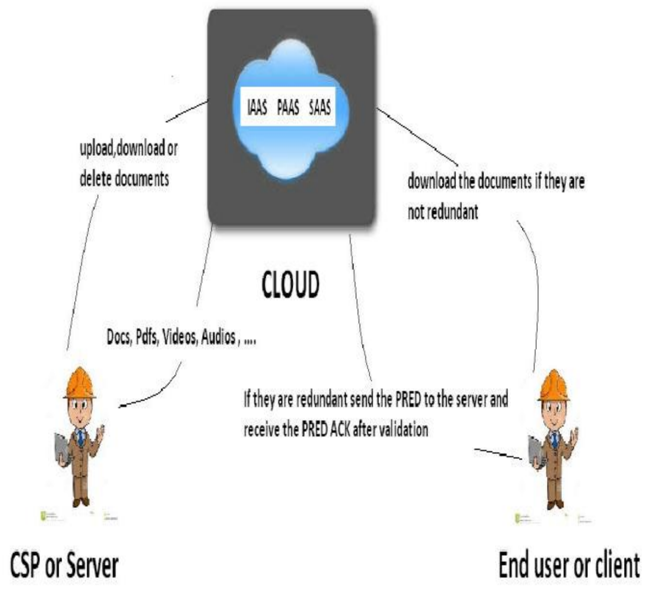

IAAS PAAS SAAS
CLOUD
upload,download or delete documents
download the documents if they are not redundant
Docs, Pdfs, Videos, Audios , ...
If they are redundant send the PRED to the server and receive the PRED ACK after validation
CSP or Server
End user or client

Fig 1: The Above Architecture Explains About the Server and Client Management of the Resources on the Cloud.

The Server registers itself with CSP for uploading, downloading and deleting Resources on the cloud. The Client after Registration can only download the Resources from the Cloud after proper validation. If the Resource is already downloaded immediately it will send a PRED to the Server and the Server replies with a PRED – ACK.

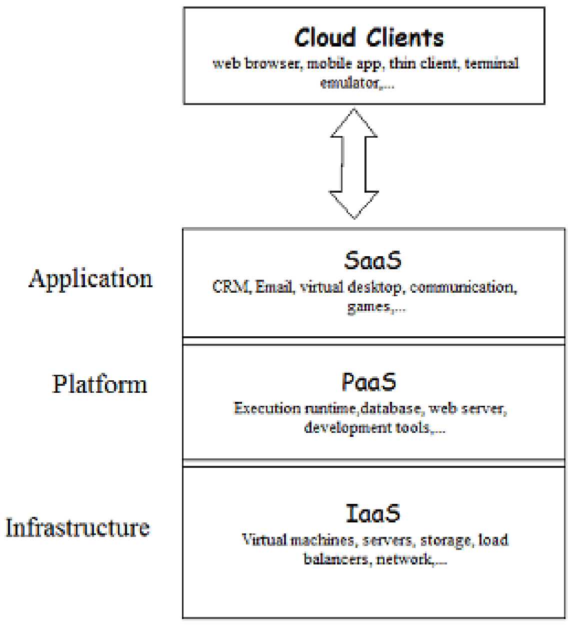

Cloud Clients
web browser, mobile app, thin client, terminal emulator,...
Application
SaaS
CRM, Email, virtual desktop, communication, games,...
Platform
PaaS
Execution runtime,database, web server, development tools,...
Infrastructure
IaaS
Virtual machines, servers, storage, load balancers, network,...

Fig 2: Service models

Cloud computing allows application software to be operated using internet-enabled devices. Clouds can be classified as public, private, and hybrid.

➢ **Service Models**

Cloud computing providers offer their services according to several fundamental models:

❖ Infrastructure as a Service(IaaS)

❖ Platform as a Service(PaaS)

❖ Software as a Service(SaaS)

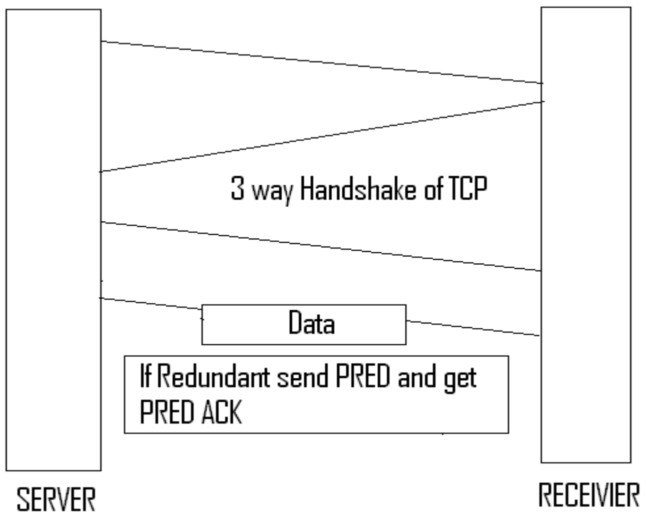


Fig 3: First The Above Diagram Establishes A 3-Way Handshake Algorithm for Establishing The Connection Between the Client and the Server

The above scenario also explains the concept of sending the PRED message and receiving PRED-ACK in the Case of redundant copy.

TCP enables two hosts to establish a connection and exchange streams of data. **TCP** guarantees delivery of data and also guarantees that packets will be delivered in the same order in which they were sent.

➢ **Connection Establishment**

To establish a connection, TCP uses a three-way handshake. Before a client attempts to connect with a server, the server must first bind to and listen at a port to open it up for connections: this is called a passive open. Once the passive open is established, a client may initiate an active open. To establish a connection, the three-way (or 3-step) handshake occurs:

1. **SYN**: The active open is performed by the client sending a SYN to the server. The client sets the segment's sequence number to a random value A.
2. **SYN-ACK**: In response, the server replies with a SYN-ACK. The acknowledgment number is set to one more than the received sequence number i.e. A+1, and the sequence number that the server chooses for the packet is another random number, B.
3. **ACK**: Finally, the client sends an ACK back to the server. The sequence number is set to the received acknowledgement value i.e. A+1, and the acknowledgement number is set to one more than the received sequence number i.e. B+1.

V. ABBREVIATIONS AND ACRONYMS

Table 1: List of Acronyms

ABBREVATIONS	
PRED	Prediction
PRED-ACK	Predictive Acknowledgement
RRE	Redundant Resource Elimination Protocol
TCP	Transmission Control Protocol
CSP	Cryptographic Service Provider

VI. UPDATED METHOD AND ALGORITHM

A Lightweight solution for all the above problems can be suggested by shifting the complete idea towards the client side with comparing the already received packets signature with the newly arrived packets. This idea finds duplicate resource before their download which will resolve all the above problems like Traffic Optimization, End users Effort and Time saving , cost will be reduced , Cloud Resources optimization can be happened, Bandwidth Consumption is reduced .

The basic idea is to download only a portion of the resource which has to be downloaded, compute its signature and find the hash code and compare it will the hash codes already stored on the store. If there is a match immediately respond the CSP with the message that a resource has been replicated. Then server can acknowledge the client about the message and can delete the duplicated file from the store such that all the resources on the cloud could be genuine and reliable.

RRE Algorithm

Server Side Algorithm

1. Server Registers with the Cloud Services
2. Server places all its resources on the Cloud
3. Server has the permission of inserting , deleting and downloading any resource

Client Side Algorithm:

1. Client Registers with the Cloud Services
2. After Proper validation the client to allowed to see and download all the resources.
3. If a resource is downloaded, Hash code is generated using SHA Algorithm and stored on the Store.
4. If a new resource is downloaded, as usually HASH CODE will be generated and compared with the already stored codes, if there is a match send a prediction to the server.

After receiving the prediction from the server an acknowledgement will be sent to the Client. The Server now can check the data in the store and delete the redundant ones there by optimizing the resources on the Cloud. A very little overhead is there on the server here where the complete idea is implemented on the client side only, so the lightweight solution of detecting and eliminating the resources on the cloud is possible.

The **hash** function is perfect, as it maps each input to a distinct **hash** value. The **meaning** of "small enough" depends on the size of the type that is used as the **hashed** value. For example, in Java, the **hash code** is a 32-bit integer.

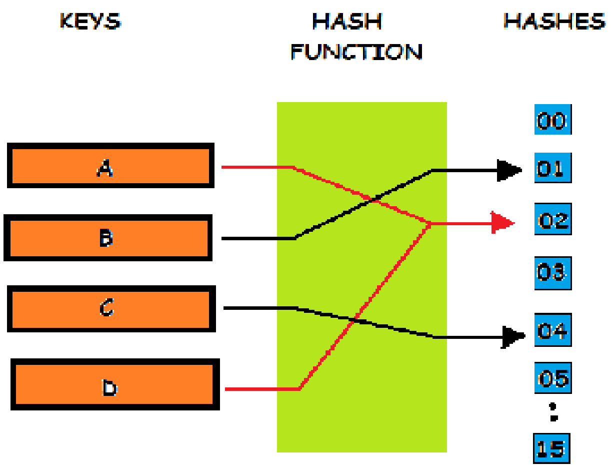
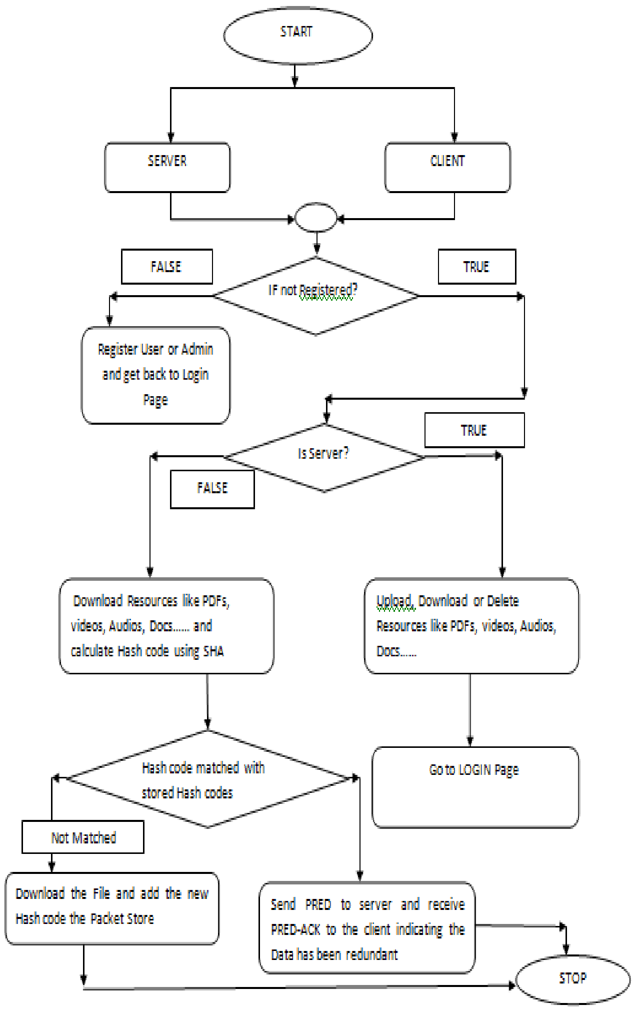


Fig 4: A Hash Function that Maps Letters to Integers From 0 To 15. There is a Collision Between Keys "A" And "D".

The **Secure Hash Algorithm** is a family of cryptographic hash functions including SHA-0, SHA-1, SHA-2, and SHA-3.

VII. FLOW CHART


START
SERVER
CLIENT
FALSE
TRUE
IF not Registered?
Register User or Admin and get back to Login Page
TRUE
Is Server?
FALSE
Download Resources like PDFs, videos, Audios, Docs...... and calculate Hash code using SHA
Upload, Download or Delete Resources like PDFs, videos, Audios, Docs......
Hash code matched with stored Hash codes
Go to LOGIN Page
Not Matched
Download the File and add the new Hash code the Packet Store
Send PRED to server and receive PRED-ACK to the client indicating the Data has been redundant
STOP

The above flowchart depicts a complete flow of control between the server and the client for navigating through all the transactions that are happened on the cloud – here our discussion will come under Resources as a service with optimization of resources, cloud mobility that includes both server and client side migration, discusses how the bandwidth can be optimized, how redundant resources on the cloud can be eliminated through our PRED and PRED-ACK concept. Further our discussions can be applied even to UDP and other protocols and domains.

VIII. RESULT

This novel idea is proven to be the best when we compare all the current solutions for achieving this. The first result we came through is Server is offloaded completely of checking for duplicate resources on the Cloud. When compared to WANAX the average packet size is extended to very large such that in some cases the size can even be unlimited, where as in the earlier the size can be multiple but not unlimited. When compared to Rabin Fingerprinting the client needs to check the complete data packets where as our idea is to check the current packet hash code with the previously occurred hash codes. When compared to Endre Chunk Match the size of the Packet is very limited that is between 32-64 Bytes.

IX. CONCLUSION

Cloud Computing offers better services by extending the idea to all the protocols such and TCP , UDP etc…. and to all the domains such as WEB APPS, MOBILE APPS, Enterprise Apps.
A Novel optimistic idea for eliminating replicated resources over the Cloud. Sometimes same videos have the same content when they have different qualities of VIDEO and AUDIO especially with videos, or even with documents. Sometimes the starting portions of several videos might be the same which has unique content somewhere after playing some time. In such cases rather than reading first part of chunk of the data it's better to read in the middle of the Document or video for calculating the HASH CODE.

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] N. T. Spring and D. Wetherall, ―A protocol-independent technique for eliminating redundant network traffic, in Proc. SIGCOMM, 2000, vol.30, pp. 87–95.

[3] A.Muthitacharoen, B. Chen, and D. Mazières, ―A low-bandwidth network file system, inProc. SOSP, 2001, pp. 174–187.

[4] Katrina LaCurts, Jeffrey C. Mogul, Hari Balakrishnan, and Yoshio Turner, ―Cicada: Predictive Gaurantees for cloud network bandwidth, MIT-CSAIL-TR-2014-004, March 24, 2014

[5] 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.

[6] Suresh Chougala and Sharavana K, Design of traffic redundancy and elimination approach for reducing cloud bandwidth and cost, IJIRCCE, Vol. 2, issue 2, February 2014.

[7] Xaunran Zong, ―Predicting application performance in the cloud, Master's thesis, Duke University, 2011.

[8] Rushil Dave, ―Mobile virtual network operator systems on cloud: An architectural and cost-benefit study, Master's thesis, august 8, 2011.

[9] A.Flint,"The next workplace revolution," Nov. 2012 [Online].Available:http://m.theatlanticcities.com/jobs-and-economy /2012/11/nextworkplace-revolution/3904/

[10] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, "Redundancy in network traffic: Findings and implications," in Proc. SIGMETRICS, 2009, pp. 37–48.

[11] 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.

[12] "PACK source code," 2011 [Online]. Available: http://www.eyalzo.com/projects/pack

[13] 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.

[14] S. Schleimer, D. S. Wilkerson, and A. Aiken, "Winnowing: Local algorithms for document fingerprinting," in Proc. SIGMOD, 2003, pp.76–85.

[15] S. Ihm, K. Park, and V. Pai, “Wide-area network acceleration for the developing world,” in Proc. USENIX ATC, 2010, pp. 18–18.

[16] H. Abu-Libdeh, L. Princehouse, and H.Weatherspoon, “RACS: A case for cloud storage diversity,” in Proc. SOCC, 2010, pp. 229–240.

[17] D. Hansen, “GMail filesystem over FUSE,” [Online]. Available: http://sr71.net/projects/gmailfs/