

Optimizing Cloud Bandwidth and Cost using Content Aware Chunk Prediction to Improve Efficiency of Retrieval

R. Josephine Grace Rexilla* and P. Shanthi

School of Computing, SASTRA University, Thanjavur – 613401, Tamil Nadu, India;
rexilla2499@gmail.com, shanthip@cse.sastra.edu

Abstract

Bandwidth, an important most demanding computing resource is mainly focused on backup service in cloud computing technology. In addition to backup solutions, communication and transferring of data also involves a considerable amount of bandwidth for efficient throughput. Management of wide area network bandwidth provision and cost is a major challenge in recent periods. Due to enormous content in cloud, cost effective bandwidth provisioning solution should be adopted by the cloud providers for better user satisfaction. Here we use a min chunk hashing technique based on the content for preserving all chunks in order to reduce the overall transfer time for individuals as well as to reduce the resource allocation load on the cloud servers. Thus, content based chunking along with the metadata considerably reduces the cost of bandwidth utilization for individuals. The bandwidth usage is predicted using cloudsim simulator through BwProvisioner class provided in the cloudbus package. A sample of large text files with redundant data is refined, chunked and hashed using Message Digest⁵ to measure the transfer time and cost performance metrics for experimental values in a dynamic work load environment.

Keywords: BwProvisioner, Chunk prediction, CloudSim, Message Digest 5, Min Chunk Hashing

1. Introduction

Cloud Computing is a paradigm where end users or customers store and retrieve their scalable resources such as files, data, software and hardware resources. It is a pay-on-demand technique where the end users access these resources via the web browsers. In order to increase their work speed, the end users adopt various traffic reduction techniques.

Traffic usually occurs due to frequent access by end users. Same files and data are cached continuously and consistently. This leads to huge traffic in the network, which increases the bandwidth and cost. Latency is the excess or delayed time period for a data to be transferred across the network. Though cloud services provide pay-as-you-use model repeated accessing of same or similar data directly increases the cost per minute which causes huge cloud costs for the customers.

Many traffic elimination techniques have been used to avoid or completely eliminate redundant traffic by comparing the incoming and outgoing data, secured signature mechanisms and several matching algorithms. In companies and public enterprises they had adopted middle-box techniques. It is a paid proprietary state protocol deployed at various internet and intranet entry and exit points. This subsequently identifies and discards the repeated data and therefore reduces the end users bandwidth cost. It cannot provide a solution for single or private user costs.

On-demand workplaces and initial investment requirements led to the rise of cloud computing in pervasive areas. The proprietary middle box technique is highly efficient only for fixed point network solutions. Mobile networks cannot benefit from such a mechanism. This led to the Content Distribution Network solution (CDN) that provides complete synchronization between the users. This drastically achieves the load balancing in

*Author for correspondence

cloud which in turn optimizes the power consumption; therefore a significant reduction in cloud cost occurs.

Sender based Traffic Redundancy Elimination (TRE) technique cannot obtain full synchronization thereby degrading the cloud efficiency. Our proposed receiver based TRE solution observes each individual incoming data and maintains a metadata for all locally cached files. It performs a hint match approach so traffic along the sender side is reduced. Synchronization is maintained by transmitting prediction acknowledgement (ACK's) of the transferred data instead of sending the repeated data. Off loading the computational work from servers to end users achieves tremendous load distribution since each user access their own respective data. This enhances cloud elasticity and mobile clouds come into existence. Hybrid approach is used when an IP address conflict occurs. This approach is effectively experimented by using the Transmission Control Protocol (TCP) option part that supports all text, audio and video streaming techniques.

Cloudsim toolkit is used to measure the performance result which is an ease of use tool to set up and simulate a cloud environment. It provides graphical outputs that depict the degree of configurability and predictability. It presents the simulation results for the overall time required to transfer the chunks and the storage of data in the cloud.

2. Related Works

Internal routing traffic due to big data¹ available in the network causes a decrease in the throughput though variable flapping timers are induced in internet routing protocol. Storage and CPU usage is unstable in large data center environment due to the over population of data within a network. NS2 simulator is used to identify the depreciation of throughput in a tiered network associated with timers that controls the routing.

In PB-PSO method², each group of tasks and resources are considered as an individual particle in particle swarm optimization technique. The tasks location and movement of allocated resources in the cloud virtual machine is considered as the position and velocity of the particle. Based on the fitness value obtained through the objective functions, the local best and global best values are computed for the task best response time and price of allocation in cloud. The computation time is reduced by updating the position and velocity for each iteration.

The energy consumed for resource allocation in cloud is done through the Bacterial Foraging Optimization (BFO) technique³ where the virtual machine position and current allocation is performed in two phases. Modified Best Fit Decreasing (MBFD) and Bacterial Foraging algorithm is used for energy conservation in cloud resource allocation. The maximum and minimum utilization of virtual machine is determined based on the random choice of virtual machines position in cloud.

The redundant chunks are predicted using both sender and receiver based approach known as hybrid approach⁴. The receiver does most of the work and when the chunks' order is interrupted, the server performs the process to preserve data integrity. It uses the anchor based sliding window to split data and signature is generated using SHA1. In some cases the client acts as the server.

The similar data present in the same file after modifications can be found using⁵. A small percentage of similarity detection among a large group of similar data uses a query file to preprocess the files using indices. It is used in the management of files, information gathering, data compressing and plagiarism detection. The tool uses a query file which contains the fingerprints of all data. Hashing, multiple pattern and tree construction techniques are used to efficiently manage the search. The input is provided as a directory and a threshold value is assigned to identify similarities and classify based on the format such as text files, binary files, etc.

Accessing files remotely over the network in⁶, involves huge latency while logging in. In LBFS, the file shared remotely consumes less bandwidth compared to other systems. It uses the SHA1 function to obtain hash values of files for transmission. The receiver accepts only new blocks that are not currently present in its cache from the server using semantics of traditional file. Shifting of offsets, increases the performance time, which is a worse case complexity.

The transmission of data between social sites, mostly occurs for same data frequently. Protocol independent similarity removal is used in middle boxes to save bandwidth at the internet service provider (ISP) links in⁷ At the internet router, an inter and intra domain redundant aware routing techniques are deployed to enhance ISP's traffic control. Thus the network routing protocols had been redesigned and achieves 10 to 50 percentages of redundancy.

SmartRE⁸, provides a best solution for traffic maintenance at the enterprise levels. The redundancy elimination

has become a network service which welcomes more bandwidth consuming applications, thereby increasing the capacity of the network. It benefits by improving the throughput and eliminating latencies making Redundancy Elimination (RE) technique more appealing. It involves the usage of the ingress node, interior node and central module where encode and decode of data occur in a slightly varying naive hop by hop approach.

The frequency and the overlapping of redundant bytes in several packets are traced over the ISP's on network edges in⁹. Traffic redundancy increases smoothly with the count of users using a particular link. This approach works on a simple First-In-First-Out (FIFO) algorithm for maintaining the packet storage and upon overflow, the early packets in the store are eliminated and hashes are freed.

WAN¹⁰ accelerator designed for compressing redundant network traffic from point-to-point communications. Multi Resolution Chunking (MRC) uses much less memory to perform intelligent load scheduling to maximize throughput while running over resource-limited platform in a mesh network environment. It provides a position and history independent content fingerprinting technique.

The data are transferred directly through the hypertext transfer protocol among peers in Napster as well as Gnutella¹¹. Such peers act as client and server based on the process. The peer content reviewing and storing are targeted towards forming a distributed storage¹² in which users will be able to go through, save, and share content in a secure and approved manner with appropriate rights.

EMFS¹³ involves the client and the mail storing cloud that presents a POSIX like file system interface allowing ubiquitous data access improving usability, scalability, and reliability. Third Party Auditor (TPA) in¹⁴, audits between the cloud provider and user for integrity. In enterprises, deduplication scheme is used as a source block level and is implemented by NetApp Data ONTAP¹⁵ and EMC Atmos¹⁶. In event coreference method¹⁷, a hybridization of sentence and feature based reduction is applied to remove duplicates. Binning¹⁸, a compact storage technique for multiple chunks provides efficient storage.

3. Proposed Work

3.1 Peer to Peer Network

A network¹⁰ comprising of several node forms a peer network in which one acts as the server and other as

clients. The links between the nodes establishes a wireless connection similar to the cloud to preserve computing resources such as storage and CPU cycles. Each node sends and receives numerous data chunks over the communication channel as shown in Figure 1. Through the activation of a communication channel in the network transferring of chunks takes place effectively. Here bandwidth acts a major constraint in the network. This creates a basis for the distribution model of nodes. We used the min chunking redundancy elimination technique for efficient scheduling in distributed systems.

3.2 Content Based Hashed Chunks

The main goal of chunking required data is to reduce the utilization of resources. Therefore the number of bytes in data compressed should be low compared to the original data. MD5 accepts a message as input and generates a fixed length output, which is comparatively less than the length of the original message. The output is referred as a hash value, fingerprint or message digests. The compressed data are further reduced by hashing value using message digest algorithms. Large and complex data can be reduced using content aware chunking scheme as in Figure 2. The hashed value computed for each chunk is used to identify the already existing data at the receiver and to sequence the retrieved chunks.

3.3 Min Chunking Redundancy Prediction

The data to be sent are split into variable size hashed chunks based on the content size and parameters. Chunks are small blocks of data containing the part of the split file. These chunks are maintained as an array of blocks on the client machine in order to save disk space. A duplication

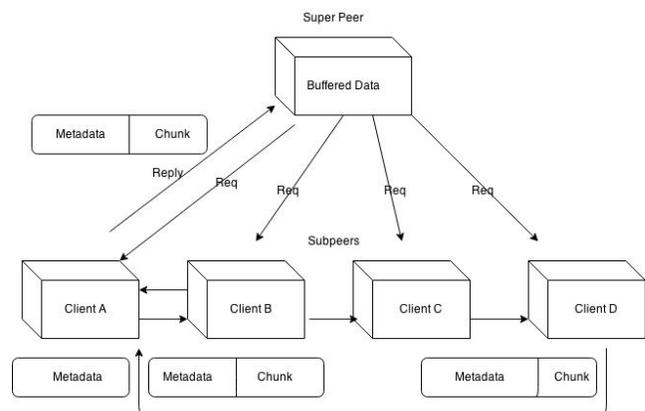


Figure 1. Min Chunking Peer to Peer Architecture.

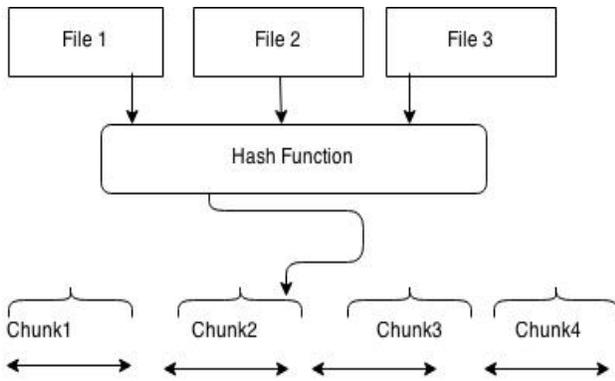


Figure 2. Splitting Hashed Chunks.

of these chunks is stored on the client before being sent to the cloud. Each duplicated chunk is matched with the newly received chunks to find the redundant points and thus avoid sending those data. The redundant free chunks are now hashed using the popular message digest MD⁵ cryptographic hashing technique using a key for both encrypting and decrypting the contents inside the blocks. The chunks are now transferred to the cloud using the TCP protocol. The chunk size is restricted and security is preserved by this hashing technique.

3.4 Chunk Metadata Store

The peer clients maintain a local storage⁴ comprising of the chunks and its metadata information. This constitutes the original file size, chunk size and the sequence number of the chunks. When a client requests for a file, it first compares the metadata of the newly incoming chunks with its locally stored data. Then it compares the first chunk of the file. If the same data has been requested, the client sends an acknowledgement for the downloading data and therefore stops the process. If not, the client accepts the incoming chunks, and store in its local storage.

Pseudo code 1: Peer Chunking Process

```

If file carries requested data then
  Evaluate chunk
  If attained chunk endpoint, then
    Activate predRecv ()
  End if
Else if PRED ACK then
  Process PredData ()
  Activate predRecv ()
End if
    
```

Pseudo code 2: predRecv ()

```

If metadata matches then
  If obtained a chunk the same as in peer storage then
    Create PREDs
    Send single ACK with PRED value
  Exit
End if
Else
  Cache chunk
End if
Forward TCP ACK alone
    
```

Pseudo code 3: PredData ()

```

For various values of PRED ACK perform
  Obtain data from peer chunk storage
  Provide data in TCP input data buffer
End for
    
```

3.5 Server Monitoring

The super peer accepts the request forwarded by the client and checks whether the same request has been sent by any other peer recently. The request alone is broadcasted to all clients in the super peer buffer. The clients in turn send an acknowledgement of the data existence. The server calculates the distance between the requested peer and the sub peers with file availability. It chooses the shortest peer among them and gives the requested peer IP address. The respective peer sends the data in the form of hashed chunks which is of a few bytes to the requested peer.

3.6 Estimate Bandwidth using Cloudsim

CloudSim is an ideal platform for modeling data centers, brokers, virtualized users and hosts. Our process is implemented on top of the cloudsim to evaluate the attained bandwidth, the time to transfer chunks and its cost as in¹³. The chunks are transferred using datagram sockets. The bandwidth allocated is obtained using get AvailableBw () and the utilized bandwidth using getUsedBw (). The cost per usage is calculated using getCostPerBw () and getCostPerStorage ().

4. Implementation

In this, we present our project implementation done using Java programming language on NetBeansIDE8.0.2

platform with the help of a cloud simulator to derive our experimental results. It runs on Windows 7 for both super peer and sub peers. At the super peer side, we use an Intel Core i3 CPU 3GHz, 2.5 GB of RAM and ATA HGST HTS545050A7 SCSI drive. The sub peers machine is based on Intel Core CPU i3 2.53GHz, 3 GB of RAM and ATA HGST HTS545050A7 SCSI drive. Our implementation uses TCP packets on both sides. We use a chunk size of minimum length according to the content in the file. Further, we predict the amount of bandwidth saved and the cost per usage across the network using CloudSim 3.0.2 simulator to obtain the accurate results.

The storage required for metadata and its respective chunk seems to be comparatively very lower than the whole file storage. This reduces the disk storage and CPU revolutions. The generation of metadata and hash value involves a less number of CPU cycles and speed. Moreover, most of the transactions occur between the sub peers which highly reduces the super peer effort except for the request and response time.

The bandwidth for each chunk of data is calculated that acquires the minimum of the allocated bandwidth compared to the normal transfer of files. This enables multiple peers to perform operations over the network, thereby eliminating traffic and congestion at the routers. The cost in turn is obviously low for each transfer of chunk benefitting the private cloud users. The results are presented in the next section for experimental purpose.

5. Results

5.1 Bandwidth

A sample set of files is saved in the peer metadata chunk store. The Table 1 shows the bandwidth consumed before and after the min chunking reduction technique. The values are obtained using the cloudsims. The Figure 3 represents the utilization of the bandwidth using bar graph.

Table 1. Bandwidth Consumption

File Size (MB)	Bandwidth consumed before min chunking	Bandwidth consumed after min chunking
5	4485	4396
6	5503	5365
22	22401	21406
28	27838	26622
77	78088	74412

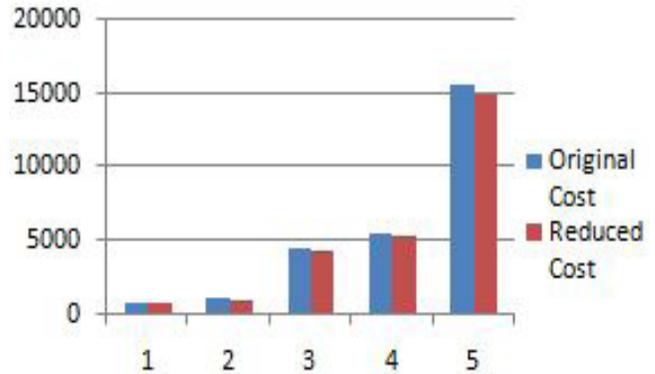


Figure 3. Consumed Bandwidth of Chunked Data Vs Original Data.

The bandwidth allocated for each data is transfer dramatically reduced to 10-15percent of the original bandwidth allocated for the respective data. This suits the cloud environment for numerous users at various periods in different locations that benefit the cloud provider. Network traffic and congestion at the routers can be greatly eliminated.

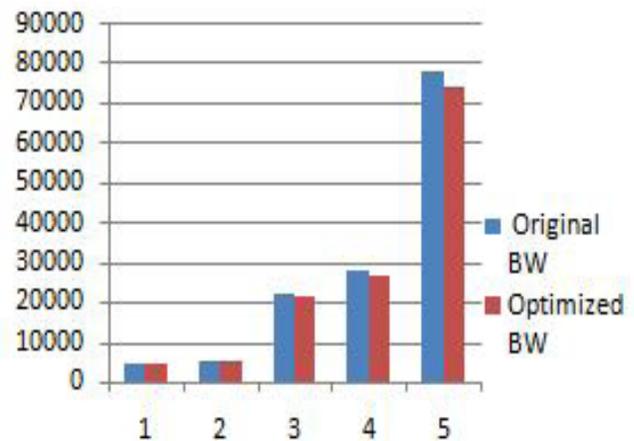


Figure 4. Cost Optimization of Chunked Data Vs Original Data.

Table 2. Reduced Cost of Optimized Bandwidth

File Size (MB)	Cost obtained before min chunking	Cost obtained after min chunking
5	897	879.2
6	1100.6	1073.0
22	4480.2	4381.2
28	5567.6	5324.4
77	15617.6	14882.4

5.2 Cost Estimation

Table 2 provides the cost of transfer per file for individual users in the cloud. This improves the cloud usage of both individuals and enterprises. It is represented using bar chart in Figure 4 where the cost reduction can be improved by 10 percent of the original cost of each transfer of data.

6. Conclusion

The pay-as-you-go model of cloud not only benefits the large enterprises or institutions, but also the individuals by the provision of low cost per use through effective min chunking redundancy elimination techniques. This technique can be further used in audio and video files that consume a major part of available bandwidth and overcome congestion in network. Various duplication techniques can be applied to optimize the consuming bandwidth.

7. References

1. Park HW, et al. Study on the impact of big data traffic caused by the unstable routing protocol. *Indian Journal of Science and Technology*. 2015; 8(S5):59–62.
2. Mohana RS. A Position balanced parallel particle swarm optimization method for resource allocation in cloud. *Indian Journal of Science and Technology*. 2015 Feb;8(S3):182–8.
3. Dhingra A, Paul S. Green cloud: heuristic based bfo technique to optimize resource allocation. *Indian Journal of Science and Technology*. 2014 May; 7(5):685–91.
4. Zohar E, Cidon I, Mokryn OO. The power of prediction: Cloud bandwidth and cost reduction. *ACM SIGCOMM Computer Communication Review*. 2011; 41(4):86–97.
5. Manber U. Finding Similar Files in a Large File System. *Usenix Winter*; 1994.
6. Muthitacharoen A, Chen B, Mazieres D, editors. A low-bandwidth network file system. *ACM SIGOPS Operating Systems Review*. 2001; 35(5):174–87.
7. Spring NT, Wetherall D. A protocol-independent technique for eliminating redundant network traffic. *ACM SIGCOMM Computer Communication Review*. 2000; 30(4):87–95.
8. Anand A, Sekar V, Akella A, editors. SmartRE: an architecture for coordinated network-wide redundancy elimination. *ACM SIGCOMM Computer Communication Review*. 2009 Oct; 39(4):87–98.
9. Gupta A, Akella A, Seshan S, Shenker S, Wang J. Understanding and exploiting network traffic redundancy. UW-Madison, Madison, WI, USA. Tech Rep.1592.2007.
10. Ihm S, Park K, Pai VS, editors. Wide-area Network Acceleration for the Developing World. *USENIX Annual Technical Conference*; 2010.
11. Krishna SS, Gummadi PK, Gribble SD. A measurement study of peer-to-peer file sharing systems. 2002
12. Androutsellis-Theotokis S, Spinellis D. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys (CSUR)*. 2004; 36(4):335–71.
13. Srinivasan J, Wei W, Ma X, Yu T. EMFS: Email-based personal cloud storage. 2011 6th IEEE International Conference on Networking, Architecture and Storage (NAS); Dalian, Liaoning; 2011. p. 248–57.
14. Mohta A, Sahu R K, Awasthi L K. Robust data security for cloud while using third party auditor. *International journal of advanced research in computer science and software engineering*. 2012; 2(2):1–5.
15. Chapman D. Tackle the Top Three VMWare Backup Challenges. *NetApp Tech OnTap*. 2008
16. Atmos Multi-tenant, distributed cloud storage for unstructured content [Online]. 2009. Available from: <http://www.emc.com/products/detail/software/atmos.htm>
17. McConky K, Nagi R, Sudit M, Hughes W. Improving event co-reference by context extraction and dynamic feature weighting. 2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA); New Orleans, LA; 2012. p. 38–43.
18. Upadhyay A, Balihalli PR, Ivaturi S, Rao S. Deduplication and compression techniques in cloud design. 2012 IEEE International Systems Conference (SysCon); 2012. p. 1–6.