International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-3, June 2015
ISSN: 2395-3470
www.ijseas.com

TIME EFFICIENT DISTRIBUTED FILE STORAGE AND SHARING USING P2P NETWORK IN CLOUD

Sapana Kapadnis[1],Prof. Ranjana Dahake[2]
Department of Computer Engineering, MET BKC, Adgoan

ABSTRACT

In cloud computing most of the application follows client/server architecture. Basic demand of cloud computing is anytime and anywhere service. Files stored at cloud can be of any type like multimedia files, virtual machine image files etc. These files are large in size as compare to other files. Proposed system is distributed file storage and sharing system which improves provisioning time, availability, scalability, speed while eliminating various bottlenecks like jitter, delay etc. Logically connected nodes share many files having large size. For improving peers limited storage space this strategy is put forward. Diversity of stored can be improve with the same storage space. This system is used to improve efficient sharing of file between peers.

Keywords: Cloud Computing, P2P system, Virtual Machine Files.

I. INTRODUCTION

The last decades have reinforced the idea that information processing and sharing can be done more efficiently centrally, on large forms of computing and storage systems accessible via the Internet. Which leads to introduction of new technology called cloud computing. Cloud computing is a technical and social reality and an emerging technology. Cloud computing provides various services like Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS)[1]. These services are metered such as user can be charged only for the resources they have. Basic reason behind lower cost in cloud computing is resource multiplexing. Data mining, gaming and social networking, scientific and engineering applications, computational financing as well many data-intensive and computational activities can be benefit from cloud computing. Various approaches are used in cloud computing like client/server approach and P2P approach. In client/server system centralized approach is followed where P2P system is highly decentralized [2].

The problem of client/server architecture is studied here by using concept of distributed file storage and sharing using P2P network in cloud. Peer-to-peer networking is a distributed application architecture which partitions tasks or workloads between peers. Equal privileges are provided to each peer. They are said to form a peer-to-peer network of nodes. Peers are both suppliers and consumers of resources, in contrast to the traditional client server model in which the consumption and supply of resources is divided. Emerging collaborative P2P systems are going beyond the era of peers doing similar things while sharing resources, and are looking for diverse peers that can bring in unique resources and capabilities to a virtual community thereby empowering it to engage in greater tasks beyond those that can be accomplished by individual peers, yet that are beneficial to all the peers[5]. In P2P networks, clients both provide and use resources. This means that unlike client-server systems, the content serving capacity of peer-to-peer networks can actually increase as more users begin to access the content [4]. This property is one of the major advantages of using P2P networks because it makes the setup and running costs very small for the original content distributor. Popularity of video streaming increasing day by day various strategies are used for video as discuss above like client/server and peer-to-peer. Same mechanism is applicable for multimedia files in case of client server approach server provides video stream through unicast links to client. As all load for broadcasting is at center server so as number of users workload on server also increases which makes this approach unscalable [3].

Here goal is to design a system which stores the file which is large in size where each peer is contributing for serving others which leads to reduction in prefetching delay. In this system chunk of file is shared among the users which can be videos or virtual machine files. A peer can access chunk or small prefix store at other peer. It reduces the workload of server. It is distributed protocol as to store and track data.

II. LITERRATURE SURVEY

A.Client/Server Architecture

In the existing system the client/server architecture is used where complete data is at server and complete workload is on single node. It is costly in terms of storage and bandwidth. It is also not scalable with amount of users[1]. In this architecture on server many clients are present and server is provider of source and client is the one who sends request for resources. Server provides service to one or more clients. Client does not bother about working of server it is only interested in response to the request [3]

B.Video-on-Demand

Video-on-Demand (VoD)[5] is a application in which we can see video any time independent of broadcast time. User has given privilege to watch/listen and select audio and video whenever the required. It does not rely on specific broadcast time. But it is costly because entire load is placed on video server. Combining this application with peer-to-peer techniques load is shifted from single node/server to peers[4].

C.GridCast

In GridCast many operation on video are provided pause, play, and seek and it also support peer sharing to improve system performance and scalability. It is live on CERNET and many videos are served by it. In high time where demand of videos is more it serves up to 23,000 users at a time. With single video caching (SVC) GridCast can decrease load on source servers by an average of 22 percent from client-server architecture. By using results of deployment of this system further improve peer sharing is done through caching and replication. But sharing is not done in efficient way[5].

D.Peercasting

In peercasting concept live stream is broken into small length files and these files are distributed by using concept of peer-to-peer sharing[6]. This concept of peercasting is used by BitTorrent in which if any user downloads any file, which belongs to swarm and may not be offline then other user from same swarm can download file instead of requesting to server, which is not done in case of GridCast[7].

E.Bit torrent

BitTorrent is used to reduce impact on server and network of distributing large files. Instead of downloading file from single node multiple peers are involved which uploads and downloads files simultaneously. First small torrent descriptor is created by user who wants to upload a file. File is available through a

BitTorrent node which at as a seed. Those who wants download file they can directly get connected to no seed or peer and can download file from peer[8].

In BitTorrent and other applications data from users devise automatically get upload without permission of user[9]. But in proposed system user has privilege of announcing data at data tracker, without permission of user file from user's storage space cannot be accessed.

III. PROPOSE SYSTEM

Concept of video-on-demand can be replaced by file-on-demand. Files in generalized form like multimedia server. Here it has been proposed that, announcer contains file where data is divided into chunk. In that case fetcher node request to announcer for specific chunk of file. Announcer node pushes the chunk to another node. After that any request related to that chunk is forwarded provider (i.e source node or other nodes under source node). All these nodes totally forms cluster of interested & it keeps growing in proportion to the number of clients needing it. So basic drawback of client server architecture is overcome by this architecture. Extra overhead on server is reduced by using peer-to-peer technology. Each peer may perform some task and contribute for file sharing and storage.Fig.1 shows system architecture for P2P File Storage and Sharing System.

As shown in fig.1 there are three components of P2P system as provider, fetcher, and announcer.

Provider/Announcer: Announcer announces the file name, client Id, hash value to data tracker. If any request for that particular ip received at data tracker will send corresponding file as per the request in chunk form to fetcher.

Datatracker: At Data Tracker announcer announce the file name, client id, hash value. Data Tracker maintain log about file and information related to provider, fetcher. Also keeps track related to each and every file. Handles request from fetcher save activities like announcement, role change etc.

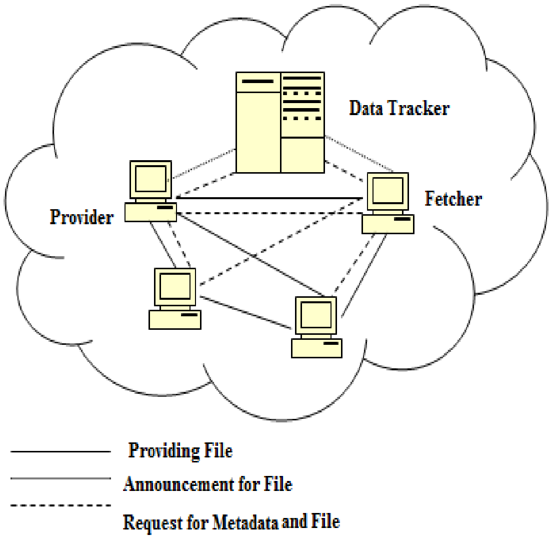

Fig.1 P2P File Storage and Sharing System

Fetcher: Fetcher requests for metadata of required file which may contain filename, file type, client id, location of provider etc. If particular file found into database client id related that file is retrieved and send to fetcher. Using client id fetcher get connects to announcer and receives file from announcer.

A.SystemFlow

1.Announcer creates a small announcement for file which he wants to distribute with other nodes. This announcement contains filename, client id of announcer as shown in step 1 of fig.2.

2. Fetcher sends request for file to data tracker along with file name.

3. Data Tracker searches metadata for that file. If any record related to that file founds data tracker sends metadata client id of provider to fetcher. If

that file is not present in database then file not found message is send to fetcher.

4. After arrival of request for file that file is divided into 1mb chunks. Each chunk is send to fetcher along with checksum which protects each and every chunk. Modification in chunk is detected which prevent malicious and accidental alteration of chunk. To achieve integrity checksum is introduced.

5. Chunks received are of same size as 1mb each. When a peer completely downloads a file, it becomes provider. This eventually shifted from peer to peer.

6. At datatracker another fetcher sends request for the file which is present with multiple nodes.

7. Data Tracker sends metadata for requested file to fetcher which contains ip address of multiple providers along with there information.

8. Fetcher will send request to these providers and as per the request some of the part of file fetcher gets downloaded from each providers.

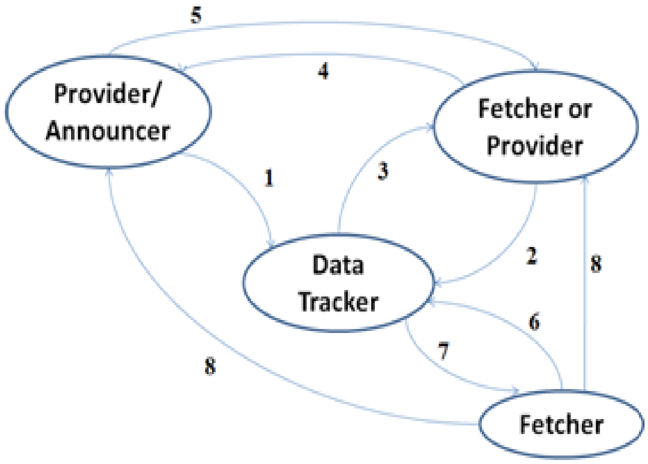

Fig.2 System Flow

IV. PROCESSING STEPS

Basic three components are involved in this system as datatracker, provider and fetcher.

Algorithm : Datatracker

Data tracker maintains metadata and displays information related to role change if any. At data tracker if any request comes from fetcher data tracker search for metadata of that file. It also displays all information related to file which is announced.

```
Definition:
uname=username
upass= password
ddate= date of download
chunks= Number of chunks of file
uploaddate=date of file upload
owner=owner of file
1. create socket
2.get ip address of data tracker
3.bind ip address
4.check for registration  of user with his uname and upass
5.WHILE TRUE
6.accept connection from provider or fetcher
7.print the role of user along with owner name
8.fetch metadata related to requested file
9.IF list contains ip address for requested file
10.display metadata at datatracker
11.IF file download is completed then change the role from fetcher to provider
12.print role of fetcher along with  time to download file, number
13. act as a fetcher
14..END WHILE
```

Algorithm : Provider and Fetcher

Any registered user can at as both provider and fetcher as per the role.

```
1.IF file announcement is equal to true
2.filename= file path
```

```
3.descr= description
4.IF file path exists
5.split the filename and open file in read mode
6.read entire content of file
7.get the length of data i.e size of input file in bytes
7.divide chunks as per the size of file
8.calculate checksum for file
9.send metadata along with announcer, filename, size, chunks
10.ELSE please enter valid file
11.ENDIF
12.IF requests are pending for file download
13.approve request of fetcher
14.IF file download equal to true then send request
14.     IF request approved
15.         download file from provider
16.     else
17.         wait for approval from provider
18.ENDIF
```

V. EXPRIMENTAL RESULTS AND ANALYSIS

Fig.3 shows time to download various types of file along with file type. As single node is involved time required to file download is more as compared to multiple nodes. All the analysis is done on the basis of results from table.1.

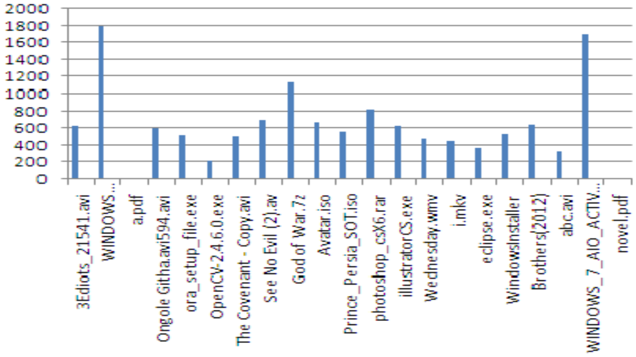

Fig.3 File Type V/S Time for Single Node

Fig.4 shows time to download various types of file along with file type. As compared to node single node is time required to file download is less. Time to download is decreases as two nodes are involved.

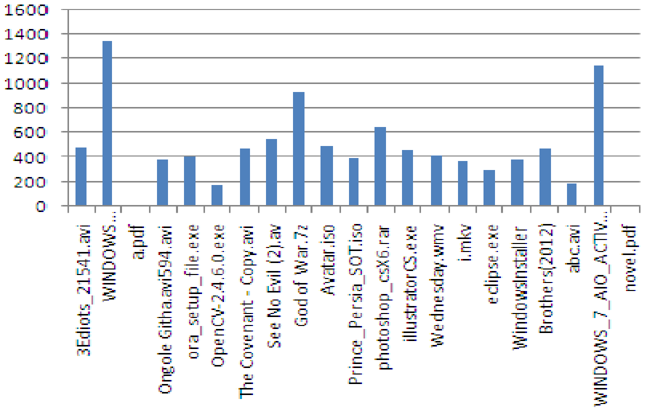

Fig.4 File Type V/S Time for Two Nodes

Fig.5 shows time to download same file from multiple nodes. As compared to graph in fig.3 and fig.4 time required for file download is less for same file. As three nodes are involved each node is contributing for file download which leads to reduction in time.

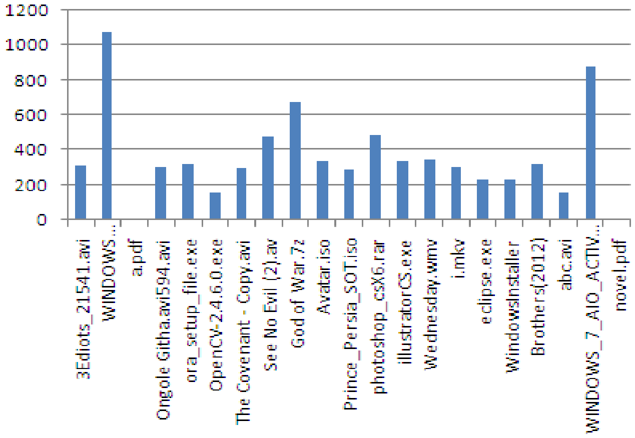

Fig.5 File Type V/S Time for Three Nodes

In fig.6 clear difference in reduction time can be observed as compared to above graphs three graph. Contribution of multiple nodes makes system more efficient and also increases system performance. Time to download the file is

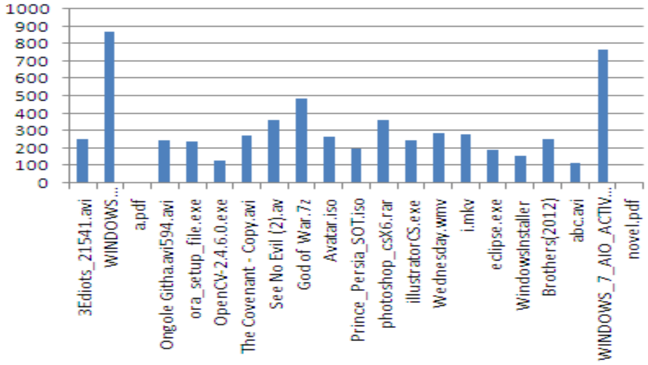

Fig.6 File Type V/S Time for Four Nodes

almost reduces by half as compared to time required for single node.

Table.1 Results

			Time In Seconds			
File name	**File Size**	**No Of Chunks**	**Time For Single Node**	**Time For 2 node**	**Time For 3 node**	**Time For 4 node**
3Ediots_21541.avi	1535MB	1535	624	480	312	250.84
WINDOWS 8_ACTIVATED.iso	3817 MB	3817	1794	1348	1074	869
a.pdf	27 MB	27	10	8	7	5
Ongole Githa.avi594.avi	1481MB	1481	593	379	303	243
ora_setup_file.exe	948MB	948	513	402	317	238
OpenCV-2.4.6.0.exe	291MB	291	221	171	149	130
The Covenant - Copy.avi	788MB	788	508	471	292	270
See No Evil (2).avi	998 MB	998	694	541	474	363
God of War.7z	2138MB	2138	1139	927	673	481
Avatar..iso	1567MB	1567	672	495	334	263
Prince_Persia_SOT.iso	1342MB	1342	563	393	287	197.34
photoshop_csX6.rar	1827MB	1827	814	639	479	363
illustratorCS.exe	1172MB	1172	623.94	461	334	247
Wednesday.wmv	752MB	752	481	415.37	341	285.41
i.mkv	700MB	700	453	374	304.1	275
eclipse.exe	526MB	526	367	296	229	192
WindowsInstaller	1220MB	1220	527	382	223	154
Brothers(2012)	845MB	845	641	472	318	251
abc.avi	780.2MB	781	322.253	180.016	149.33	115.84
WINDOWS_7_AIO_ACTIVATED.iso	3817.52 MB	3818	1694.1	1148.7	874.6	769.5
novel.pdf	13 MB	13	8.033	6.86	4.39	3.2

Form table.1 comparative analysis can be observed for file with variable size and variable type.

CONCLUSION

Distributed P2P File Storage and Sharing system stores files in a peer to peer network where each peer is contributing for serving others which leads to reduction in perfecting delay as well as eliminating bandwidth bottleneck. In this system files are shared among the nodes in a distributed way. A peer to peer approach is best suited for downloading or sharing large data files. It is distributed protocol as to store and track data while continuously verifying the data integrity using check sums. Ratio of file download from single peer to four peers is 2:1 so time almost reduces by half. It reduces workload of single node/peer also provides time efficient distribution and sharing of file having large size which reduces time of file sharing.

REFERENCES

[1] Mladen A. Vouk “Cloud Computing –Issues, Research and Implementations” Department of Computer Science, North Carolina State University, Raleigh, North Carolina, USA - CIT 16, 2008, 4, 235–246

[2] Ilango Sriram, Ali Khajeh-Hosseini ”Research Agenda in Cloud Technologies” Department of Computer Science, University of Bristol, Bristol UK

[3] IEEE COMMUNICATIONS SURVEYS TUTORIALS on "A Survey and Synthesis of User Behavior Measurements in P2P Streaming Systems" by Ihsan Ullah, Guillaume Doyen, Gregory Bonnet

[4] Z. Li, H. Shen, H.Wang, G. Liu and J. Li,"SocialTube:P2P-assisted Video Sharing in Online Social Networks," In Proc. IEEE TRANSACTIONS ON PARALLEL AND

DISTRIBUTED SYSTEMS, VOL. 25, NO. 9, SEPTEMBER 2014
[5] B. Cheng, L. Stein, H. Jin, X. Liao, and Z. Zhang, "Gridcast: Improving Peer Sharing for P2P VoD," ACM Trans. Multimedia Computer., Communication., Appl., vol. 4, no. 4, p. 26, Oct. 2008.
[6] Yaning Liu ,Gwendal Simon ,"Peer-to-Peer Time-shifted Streaming Systems ",Institute Telecom - Telecom Bretagne, France,November 6, 2009.
[7] Laizhong Cui and Nan Lu SocialStreaming: P2P-assisted Streaming in Social Networks In IEEE Trans. on Computers,2013.
[8] J.A. Pouwelse, P. Garbacki, D.H.J. Epema, H.J. Sips, THE BITTORRENT P2P FILE-SHARING SYSTEM: MEASUREMENTS AND ANALYSIS, Department of Computer Science, Delft University of Technology, the Netherlands
[9] H. Wang, J. Liu, and K. Xu, On the Locality of Bittorrent-Based Video File Swarming, in Proc. P2P, 2009, p. 12.
[10] Peersim: A Peer-to-Peer Simulator. [Online]. Available: http:// peer-sim.sourceforge.net/.
[11]PlanetLab. [Online]. Available: http://www.planet-lab.org/.
[12] Facebook Users Average 7 hrs a Month in January as Digital Universe Expands. [Online].Available:http://blog.nielsen.com/nielsenwire.