RESEARCH ARTICLE OPEN ACCESS

Enhancing Downloading Time By Using Content Distribution Algorithm

VILSA V S
Department of Computer Science and Technology
TKM Institute of Technology, Kollam, Kerala Mailid-vilsavijay@gmail.com

ABSTRACT:
A traditional model in peer-to-peer (P2P) content distribution is that of a large community of intermittently connected nodes that cooperate to share files. Because nodes are intermittently connected, the P2P community must replicate and replace files as a function of their popularity to achieve satisfactory performance. In this paper, I develop a technique to enhance the downloading time with the aid of content distribution technique in the Peer-to-Peer architecture. In this study each peer will be sending video file request to the server. Server has the capability to manage multiple requests with the help of defined number of Caches. Each video file will be partitioned into equal sized chunks and assembly information file like a torrent file in the BitTorrent application. The peer requested movie will be distributed from any of the available caches by equal sized chunks. The service mode have both server and helper mode. The Enhanced Automated Mode Selection (EAMS) algorithm will help server to select the correct service mode to serve in the peer community. This will benefit server to avail the requested file else will block the request until the availability of the file to utilize the peer bandwidth.
Key words: CDN, Cloud Downloadable, Helpers, Cloud server, File download, Peer-to-Peer, Dataset.

I. INTRODUCTION

Video content downloading became most popular in the modern world. This creates the prosperous in growth of internet usage and the Video content has become the main traffic of internet. Due to the explosive growth of the Internet and increasing demand for multimedia information on the web, video content distribution over the Internet has received tremendous attention from the community. The most common approach for such is the peer-to-peer approach. In this approach, peers who create demand for videos also share their content with other peers. The service capacity thus increases automatically with increasing peer population, making scalability an advantage of the peer-to-peer solution but a high speed file downloading is not guaranteed. So, to remedy this, a cloud downloading scheme is deployed with files as equal sized parts. The assembly of these parts will be directed through an assembly information file called link. This system used two design philosophies using cloud either as a server or a server to suit it to any operating system scenarios- the *server mode* when video population is large compared to cache size, and the *helper mode* when peer request rate is high compared to server bandwidth. In this paper, we proposed an enhanced feature which can serve multiple concurrent requests for videos having different sizes at a time. The main objectives of all video providers are to minimize their video download time by using effective techniques. Content distribution systems have traditionally adopted one of two architectures: infrastructure-based content delivery networks (CDNs), in which clients download content from dedicated, centrally managed servers, and peer-to-peer CDNs, in which clients download content from each other.

P2P content distribution mainly relies on the unstable but numerous end users to form peer to-peer data swarms, where data is directly exchanged between neighboring peers. That is, In a P2P content distribution systems, peers who create demand for videos also share their content with other peers. The service capacity thus increases automatically with increasing peer population, making scalability an advantage of the P2P solution. The real strength of P2P shows when a popular video is distributed, because a popular video is shared by a number of peers and more peers usually imply higher data health and higher degree of download parallelism, which further lead to higher data transfer rate. As to an unpopular video, it is often difficult to find a corresponding peer swarm. Even if the peer swarm exists, the few peers are unlikely to have high data health or high data transfer rate, and thus each peer has to stay online for long hours to wait for the download completion — a tedious process of low energy efficiency.

In a word, although CDN and P2P generally work well in distributing popular videos, neither of them is able to provide satisfactory content distribution service for unpopular videos, due to low data health or low data transfer rate. To overcome this problem cloud downloading scheme has been deployed in P2P video downloading systems to enhance the performance of downloading. A cloud storage system is used to cache a large fraction of video content, and high bandwidth is provided to access this cache. Peers can get a big performance boost by connecting to cloud downloading.

II. RELATED WORK

The main techniques used for video content distribution has been the CDN and later peer to peer approach. The P2P architecture has been developed in order to address the problem of bandwidth, storage, computation capability and unpredictable connectivity etc of mobile devices as in [5].Then in paper [6] an efficient mechanism for Content Distribution in a Peer-to-Peer (P2P) Network has been proposed in which the to-be-distributed content is splits into many small blocks, so that more resourceful nodes may redistribute more blocks, and less resourceful nodes may redistribute less blocks. Most of these works up to now was based on peer to peer scheme only. But this scheme poses certain performance issues. So recently, with the advent of cloud computing most researchers are trying to implement concept of cloud on video content distribution. And this was successfully implemented in paper [6].In this paper**,** Cloud Utilities was used to achieve high-quality Content Distribution for unpopular videos.

III. CLOUD DOWNLOADING SCHEME

The main approach used in this paper is the adaptive cloud download scheme which was proposed in paper [1]. The principle of cloud download scheme is as follows: Primarily, a peer sends his video request to the cloud. The video request contains a file link which can be an HTTP/FTP link, a BitTorrent [2] link. Subsequently, the cloud downloads the requested video from the file link and stores it in the cloud cache. Then the cloud notifies the peer and the peer can usually retrieve his requested video (whether popular or unpopular) from the cloud with high data rate in any place at any time. In practice, the cloud does not need to notify the peer when his requested video is available. Instead, the peer actively checks the download progress itself and takes corresponding actions. The main advantage of this approach is that Cloud download uses the cloud server to download the requested videos, so their peer group does not need to be online all the time.

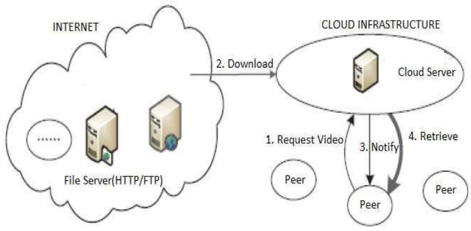

Figure 1: Cloud Download Architecture

IV. PROPOSED SYSTEM

The proposed system design aims at improving the efficiency of the video downloading service in peer to peer network with cloud storage as backup. Also, a multiple concurrent request model has been proposed. The system also has multiple cache strategy schemes to manage the cache bottleneck. This proposed system also allows a peer to download concurrent videos of different sizes so that constraint regarding size is overcome. In this system, a cloud storage system is used to cache a large fraction of video content, and high bandwidth is provided to access this caches. Peers can get a big performance boost by connecting to cloud downloading The current system provide high efficiency of video downloading service in cloud peer-peer network. The cloud server, which act as backup, operate in two service modes - as a helper or server depending on the situation.

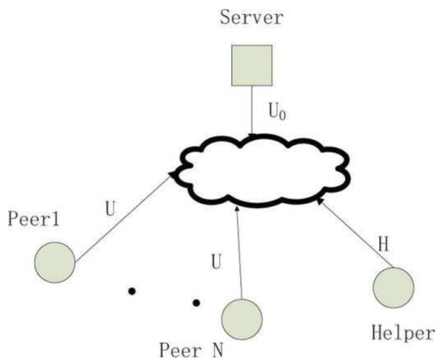

Figure 3: Movie download with Helper

In the helper mode, the cloud server begins to help the downloading request without caching the video in its cloud storage. The cloud server downloads video chunks from the P2P system [6] and redistributes it to other peers who are requested these chunks. Then, these chunks are discarded while in Server Mode the downloading requests are not served until the requested video is cached by the cloud storage. The requests for videos not in the caches are blocked. The cloud storage is updated periodically in order to serve blocked request.

V. ASSUMPTIONS

- The set of videos remains unchanged. The study of video population and popularity churn is for future work.
- Peer population is much larger than video population.
- All videos have different size. The cloud storage can only store a small subset of the videos.
- Peers have the same upload capacity.
- A multiple concurrent request model is formulated.
- Peers are fully connected, forming a full mesh topology. [7]
- The cloud server is able to replace any cached video with a new one instantly. In other words, it assumes the time for video replacement can be ignored.
- The video size is large enough to be divided into an infinite number of small chunks.

VI. PROPOSED SYSTEM DESIGN

The proposed system is divided into five modules: 1. Cloud and Peer-to-Peer Network Configuration, 2. Movie upload and segmentation, 3. Multiple Cache Strategy, 4. Implementation of enhanced automated mode selection algorithm, and 5. Chunk distribution and File Integration.

1. Cloud and Peer-to-Peer Configuration: Cloud computing is described as a subset of grid computing concerned with the use of special shared computing resources. For this reason it is described as a hybrid model exploiting computer networks resources, chiefly Internet, enhancing the features of the client/server scheme. From a sociological standpoint on the other hand, by delocalizing hardware and software resources cloud computing changes the way the user works as he/she has to interact with the "clouds" on-line, instead of in the traditional stand-alone mode. In the first module I am going to configure the peer-peer network and the cloud. The proposed system consists of N number of peers.

2. Movie upload and segmentation: Peers will upload the video to the cloud server explicitly. The video is split into different chunks. Chunks are identified by a progressive number assigned by the content server after segmentation. The chunks integration details are encoded in the assembly information file called link. This will be the first part send to upload. The size of chunk is a key factor in the system and thus should be chosen properly. Every peer will lie in an overlay network. Each peer is aware of the network structure.

3. Multiple Cache Strategy: The Cloud Server will have multiple caches to ease the file request process. The cloud server will receive download requests from each peer. The cache allocation strategy is based on the status of each cache. In this proposed system I have defined 3 cache storages to server the request. Each cache will have different set of datasets. The datasets will be refreshed from the cloud server periodically. Each cache will have two statuses as either ‘idle’ or ‘busy’. ‘Idle’ status refers the cache is not allocated to process any request and the ‘Busy’ status refers the cache is allocated to serve a request.

4. *Implementation of enhanced automated mode selection algorithm:* In this module is to implement the two modes by the cloud server namely the server mode and the helper mode. And will implement the Enhanced Adaptive Mode Selection Algorithm (EAMS) based upon that the server will adaptive to the environment and based on that automatically switching the mode to increase the throughput. The EAMS is an enhanced version of the AMS algorithm implemented in the base paper. AMS [1] switches service mode from server mode to helper mode when too many peers request blocked movies, and vice versa. The ability of AMS to achieve good performance in different operating regimes is validated by simulation.

EAMS is an adaptive algorithm which used to determine the service mode, either server mode or helpers mode. There are both strengths and drawbacks for both the helper mode and server mode. The helper mode wastes P2P resource because the cloud server needs to keep downloading new content to help peers; while the server mode wastes the bandwidth resource of blocked peers. The cloud server adjusts its strategy periodically, by running the following Enhanced Automatic Mode Selection (AMS) algorithm to determine the mode for each movie. We assume the value of N1 known. The movies in helper mode have higher priority to be included into cloud storage. Then, we consider the other movies in the order of decreasing peer population.

As discussed [1], in server mode, any request for a video not cached is blocked until the cloud storage update. Therefore, it is necessary to differentiate peers as downloading peers and waiting peers. The former are peers downloading a video cached by the cloud storage, whereas the latter are peers waiting for the cloud storage to be refreshed. In helper mode, the cloud server serves all downloading requests whether the video is cached or not. If the video is not cached, the cloud server will relay and amplify the video content downloaded from other peers.

As per analysis the weakness of helper mode is the additional bandwidth cost to download the requested video by cloud server. The benefit is that more peers can contribute their upload capacity by switching their state from waiting to downloading. Thus Algorithm compares the cost and the benefit and start helper mode once the benefit is larger than cost.

Where,

κ is the subset of videos cached by cloud storage, Nj is the peer population of movie j, U is the peer upload capacity, H is the total upload capacity of cloud server(s) N1 is the expected population of peer served by the cloud, αj is the probability that the cloud server can help any peer watching movie j by caching only one particular chunk and K is the number of cache videos. Cu refers to Cache and I indicate the cache number.

5. Chunk distribution and File Integration:

The equal sized chunks will be distributed to peers upon request. The link file will transfer in the beginning to collect the assembly information. Once the peer has received the chunks will start integration based on the link file information. The final movie file will be the integration of all chunks.

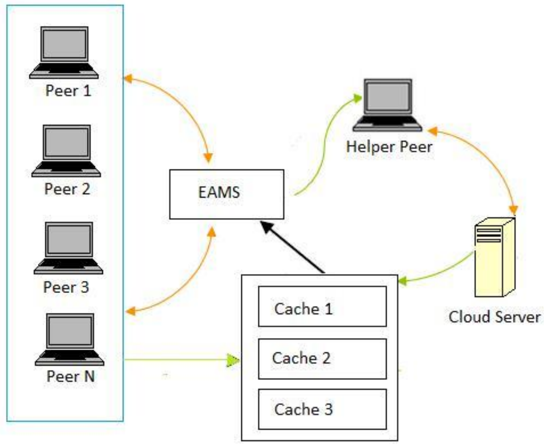

Figure 2: Implementation of Service Modes

Algorithm: Enhanced Automatic Mode Selection (EAMS) Algorithm

1: **for** each movie j not in κ. **do**
2: **if** the active movie is less than K **then**

3: Update cloud storage to add movie by replacing any movie without request.
4: N1 = N1 + Nj
5: **else**
6: **if** (H / N1 + αj Nj) < Nj U **then**
7: Use helper mode for movie j
8: if (Cu[i] == 'IDLE') then
9: Serve the request movie j
10: else Cu[i]=='BUSY' Then
11: Request will be hold until Cu[i] == 'IDLE'
8: N1 = N1 + αjNj
9: **else**

10: Keep blocking peers requesting for movie j as server mode
11: **end if**
12: **end if**
13: **end for**

VII. CONCLUSIONS

In this work I discussed the various approaches of a cloud downloading scheme. It was found that the service mode was found suitable in different scenarios. Helper mode wastes some server bandwidth, but is best at leveraging P2P capacity when request load is high. On the other hand, server mode is most efficient for dealing with large video population relative to the cache size. So, one of these two service mode was chosen using EAMS. By enabling the system to make concurrent video request at a time, the efficiency of the system was also increased. And the restriction on video size is also removed.

Video content distribution is becoming a kill application of the Internet owing to the users' inflating requirements on both video quantity and quality, so it is of great importance to investigate how to achieve high-quality content distribution for both popular and unpopular videos. My study provides practical experiences and valuable heuristics for making use of cloud utilities to achieve efficient Internet services.

REFERENCES

[1] An Adaptive Cloud Downloading Service, Yipeng Zhou*, Member, IEEE*, Tom Z. J. Fu, Dah Ming Chiu*, Fellow, IEEE*, and Yan Huang
[2] Y. Huang, Z. Li, G. Liu, and Y. Dai, "Cloud download: Using cloud utilities to achieve high-quality content distribution for unpopular videos," in *Proc. ACM Multimedia*, 2011, pp. 213–222.
[3] TheBitTorrentprotocolspecification .http://ww w.BitTorrent.org/beps/bep 0003.html.
[4] PPLive web site. http://www.pptv.com
[5] J. Cheng, Y. Li, L. Jiao, and J. Ma, ―A new mobile peer-to-peer architecture,‖ in *Proc. 5th WSEAS Int. Conf. Appl. Comput. Sci.*, 2006, pp. 1083–1088.
[6] J. Li, P. A. Chou, and C. Zhang, ―Mutual cast: An efficient mechanism for content distribution in a p2p network, ‖ Microsoft Research, MSR-TR-2004, 2005
[7] B. Fan, J. C. Lui, and D. M. Chiu, ―The delicate tradeoffs in BitTorrent like file sharing protocol design, ‖ in *Proc. ICNP*, 2006, pp. 239–248.