IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET)
ISSN(E): 2321-8843; ISSN(P): 2347-4599
Vol. 2, Issue 4, Apr 2014, 63-68
© Impact Journals

AUTOMATED AND ADAPTIVE DOWNLOAD SERVICE USING P2P APPROACH IN CLOUD

T. MARIA AROCKIA SYLVIA & P. AKILANDESWARI

PG Student, Department of Computer Science & Engineering, SRM University, Chennai, Tamil Nadu, India

ABSTRACT

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 academic and industry. 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 having a high service capacity using eye Os. This system used two design philosophies using cloud either as a server or a server to suit itself 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. We design an adaptive algorithm (AMS) to select the service mode automatically. The ability of AMS to achieve good performance in different operating regimes is validated by simulation.

KEYWORDS: Cloud Server, File Downloading, Helper, Peer-to Peer, Video

INTRODUCTION

Today, with the growth of internet and information technology there has been considerable increase in the trends of video downloading. The most common approach for such is the peer-peer. In a Peer-Peer content distribution system, 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-Peer solution. The real strength of Peer-Peer shows when a popular video is downloaded, because a popular video is downloaded, because a popular video is shared by a number of peers and more peers usually imply higher data rate and higher degree of download parallelism, which further lead to higher data transfer rate. Peers requesting unpopular videos often suffer low downloading rate. To remedy this Cloud environment is created using eye Os that seeks to enable collaboration and communication among users. It is a private-cloud application platform with a web-based desktop interface. First, a user sends his video request to the cloud server. Videos are splitted into different chunks by using "ffmpeg". So far, people can use -ss and -t to cut the video chunk out of the input video. But if they want to cut the video to multiple output chunks, they would have to start ffmpeg several times. Then the splitted video is stored in the server. There are two generic service modes for cloud servers. In the first mode, the cloud server is primarily focused on serving the content already cached at the cloud storage system. Requests for content not in the cache are blocked until such content becomes cached. The cloud storage system updates its cache periodically to replace content without requests by content with requests waiting. We call this the server mode. An alternative mode is the helper mode, in which the cloud server does not block any requests. For videos that are not cached, the cloud server simply relay chunks from some peers to other peers, acting as a helper peer. The Cloud server adjusts its strategy periodically by using AMS algorithm.

Helper mode wastes some server bandwidth, but it is best suited, 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 AMS. 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.

RELATED WORK

- **Peer-Peer Architecture**

A peer-to-peer (P2P) network is a type of decentralized and distributed network architecture in which individual nodes in the network (called *"peers"*) act as both suppliers and consumers of resources. In a peer-to-peer network, tasks (such as searching for files or streaming audio/video) are shared amongst multiple interconnected peers who each make a portion of their resources (such as processing power, disk storage or network bandwidth) directly available to other network participants, without the need for centralized coordination by servers. The architecture is shown in figure 1.

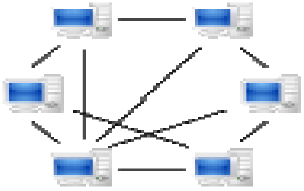

Figure 1: Peer-Peer Architecture

- **Cloud Environment Using Eye OS**

A cloud OS simply refers to an operating system (or an interface filled with a complete suite of desktop applications) that resides on the Web and you can access to it anytime, anywhere as long as you have an Internet connection. While there are plenty of cloud OS out there that you can sign up and use for free, there might be instances where you want to have your own dedicated cloud OS. First of all, signing up a free account with third-party cloud OS often means that you have limited file storage space and all your data are stored in other people’s server. Next, the connection speed is dependent on the number of active users at any time. The more popular the site is, the slower it will get when you are using it. If what you want is your own dedicated Web OS that you can use to manage your online stuff, and also to provide an Environment to collaborate with your colleagues/partners, then we can create cloud environment using eye OS. So high speed video downloading is achieved using cloud downloading with eye Os. The architecture is shown in figure 2.

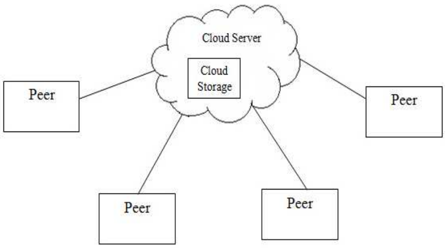

Figure 2: Cloud and Peer Network

CLOUD AND PEER TO PEER NETWORK

Cloud computing has computational and sociological implications. In computational terms 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. Peers will upload the video to the cloud server. Videos are splitted into different chunks by using "ffmpeg". Chunks are identified by a progressive number assigned by the content server. Peers organize in an overlay network and blocks are exchanged among neighbor peers. The size of chunk is a key factor in the system and thus should be chosen properly. In order to get multiple output chunks efficiently we use ffmpeg method.

CLOUD DOWNLOAD SCHEME

We are going to implement the two modes by the Cloud server namely the server mode and the helper mode. And we will implement the Adaptive mode selection algorithm(AMS)based upon that the server will adaptive to the environment and based on that automatically switching the mode to increase the efficiency /throughput

- **Helper Mode**

In helper mode cloud server download the entire request whether video is cached or not. Then it redistributes to other peer who are without these chunks.

- **Server Mode**

In server mode, any request for a video not cached is blocked until the cloud storage update. Therefore we have downloading peer and waiting peer. The downloading peer will Cache the video in the cloud storage whereas the waiting peer has to wait for the cloud storage to get updated. Architecture is shown in figure 3

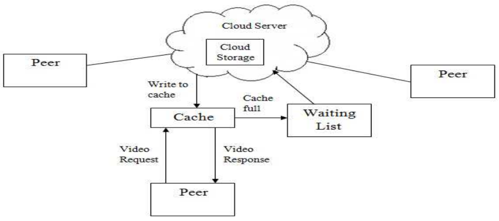

Figure 3: Helper Mode and Server Mode

- **AMS Algorithm**

By seeing the strategy, the cloud server has to decide which mode to select based on AMS algorithm. The helper mode wastes the bandwidth because it redistribute to other peers. The server mode wastes the bandwidth by wasting the blocked peers which is not cached in the cloud storage. The cloud server adjusts its strategy.

Periodically, by running the following Automatic Mode Selection (AMS) algorithm to determine the mode for each video. It is shown in figure 4.

The AMS Algorithm is Given Below

```
For Each movie j not in k. do
    If the active movie is less than k then
        Update cloud storage to add movie by j replacing any movie without request.
        $n^1=n^1+n^j$
    Else
        If $(h/n^1+a_jn_j)<n^ju$ then
            Use helper mode for movie j
            $n^1=n^1+a_jn_j$
        Else
            Keep blocking peers requesting for movie j
        End if
    End if
End for
```

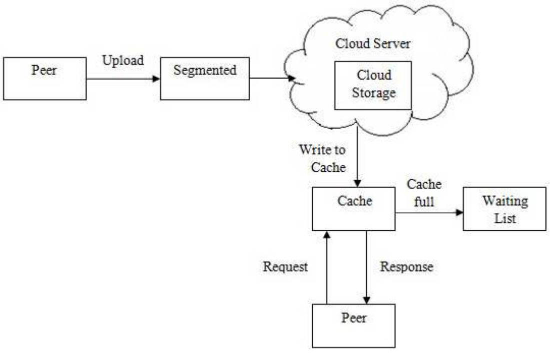

Figure 4: System Design for Video Downloading

The AMS algorithm has to decide based on the peer request. The bandwidth and efficiency is improved by selecting the mode and switching between downloading and waiting peer by using AMS algorithm.

EVALUATION

- **Throughput under Download Bandwidth**

The above sections assume that the only bottleneck in video downloading is the upload bandwidths of the peer nodes. Consider a peer node *i* with upload bandwidth B*ui*.. Let its link bandwidth to the content-receiving peer node *j*

The Effective Upload Bandwidth of Node *I* Becomes

$B^ui=M\ MINB_{ij}$

This effective upload bandwidth can be used in equation (1) to obtain the new throughput. When a content-receiving peer node has download bandwidth less than the throughput given in equation (1) (which is based only on the upload bandwidths), such a node will also be a bottleneck of video downloading. In such scenario, the overall throughput will be the minimum download bandwidth of all the content-receiving peer nodes. This is because all nodes have to wait for the slowest node to finish before they can resume delivery.

CONCLUSIONS AND FUTURE WORK

The increasing demand in the multimedia paved a way for the efficient video downloading so we implemented using cloud server. It was found that the service model 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 AMS. 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

REFERENCES

1. 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.
2. A [Online]. Available: http://www.akamai.com
3. PP Live. [Online]. Available: http://www.pptv.com/
4. PP Stream. [Online]. Available: http://www.ppstream.com/
5. Thunder. [Online]. Available: http://dl.xunlei.com/xl7.html
6. QQ. [Online]. Available: http://xf.qq.com/
7. H.-C. Hsiao and C.-T. King, “Bristle: A mobile structured peer-to-peer architecture,” in *Proc. Int. Parallel Distrib Process. Symp.*, 2003.
8. 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.
9. F.-U. Andersen, H. d Meer, I. Dedinski, C. Kappler, A. Mader, J. Oberender, and K. Tutschku, “An architecture concept for mobile p2p file sharing services,” in *Proc. Workshop Informatik—Algorithms and Protocols for Efficient Peer-to-Peer Applications*, 2004, pp. 1–5.
10. D. Qiu and R. Srikant, “Modeling and performance analysis of bit torrent like peer to peer networks,” in *Proc. ACM Sigcomm*, 2004.
11. J. Mundinger, R. Weber, and G. Weiss, “Optimal scheduling of peer-topeer file dissemination,” *J. Scheduling*, vol. 11, no. 2, pp. 105–120, Apr. 2008.
12. D. Chiu, R. Yeung, J. Huang, and B. Fan, “Can network coding help in p2p networks,” in *Proc. 2nd Workshop Network Coding*, 2006, pp.1–5, invited paper.

13. B. Fan, J. C. Lui, and D. M. Chiu, "The delicate tradeoffs in bit torrent like file sharing protocol design," in *Proc. ICNP*, 2006, pp. 239–248.

14. 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.

15. M. Lin, B. Fan, J. C. Lui, and D.-M. Chiu, "Stochastic analysis of file swarming systems," *Performance Eval.*, vol. 64, no. 9–12, pp.856–875, 2007.