ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

An Adaptive Downloading Service from Object Storage Cloud

T.Avinash
M.Tech,
Dept of Software Engineering (SE),
Vinuthna Institute of Technology & Sciences.

T. Moulika
Asst. Professor,
Dept of Software Engineering (SE),
Vinuthna Institute of Technology & Sciences.

Abstract:

Video content downloading using P2P approach does not always give good performance though it isscalable. Subscription-based premium services, referred to as cloud downloading have emerged recently. In this service, based on user downloading requests the cloud storage and server caches user interested content, and updates the cache. The request is held in a waiting state until the cache is updated, if a requested video is not in the cache. This design is called server mode. An alternative design is to let the cloud server behaving as a helper peer, serve all downloading requests as soon as they arrive.

This design is called helper mode. The proposed model and analysis show that both of these designs are useful for certain operating regimes. The server mode is good at scaling with video population size, while the helper mode is good at handling high request rate. To design the service mode automatically, an adaptive algorithm (AMS) is proposed. When too many peers request for blocked movies, AMS intuitively switches service mode from server mode to helper mode and vice versa.

Keywords:

Object Storage, OpenStack, Swift, Cloud Server, file downloading, helper, peer-to-peer, video.

I.INTRODUCTION:

Internet traffic is dominated by video today. While watching video content, users expect smooth playback and low latency and if it is not provided vendors may lose customers, hence video content downloading from cloud is challenging. The two mechanisms for video content distribution include Content Delivery Network (CDN) and Peer-to-Peer (P2P).

CDN, a traditional solution based on deploying servers at the edge of the network, near video access points. A limitation of CDN is scalability, because when there are a large number of concurrent peer requests, the server capacity becomes a bottleneck. In P2P, peers who create demand for videos also share their content with other peers. With the increase in peer population, service capacity also increases, making it scalable. Usually peers may leave and join the system at any time, making the network unstable. Because of its high bandwidth requirement and the fast growing video population, video content distribution is a challenging problem. A dedicated server is not commonly deployed for service capacity in file sharing scenarios. Peers, those requesting unpopular videos often suffer low downloading rate. To enhance the performance of downloading, a cloud downloading service is deployed in P2P network. A large fraction of video content is cached with the use of a cloud storage system, and high bandwidth is provided to access this cache. By connecting to cloud downloading, peers can get a big performance boost. The architecture and implementation of such a cloud downloading system is introduced in [1]. This work is based on a P2P file sharing system in an educational institution with more than three thousand students. Finally, a cloud downloading system with high service capacity is necessary. The key to determine system performance is to know how to utilize the resource of cloud server efficiently. There are two generic service modes for cloud servers.

Server Mode :

In this mode, the cloud server is primarily focused on serving the content which is already cached at the cloud storage system. Requests for content which are not in cache are blocked until such content becomes cached. Periodically the cloud storage system updates the cache to replace content with requests waiting.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Helper Mode :

In this mode, the cloud server does not block any requests. The cloud server simply relay chunks from some peers to other peers, acting as a helper peer, for the videos that are not cached. A study to compare these two modes analytically says that The results are interesting, in the sense that both modes can be advantageous for some operating regimes—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 integrate these two modes into asingle adaptive cloud downloading service [2]. The benefit is that more peers can contribute their upload capacity by switching their state from waiting to downloading. Server mode is most efficient for dealing with large video population relative to the cache size.

The rest of the paper is organized as follows: Section II introduces notations and the assumptions of the analysis done in [2]. Section III discusses a static model for this simple analysis. The analysis of both helper mode and server mode in a dynamic model is included in Section IV. An automatic service mode selection algorithm proposed in [1] and its implementation is covered in Related Work as Section V. Section VI covers the results.

II.ASSUMPTIONS:

Some simplifying assumptions for the models of this paper are made to keep the problem tractable.

- The set of videos remains unchanged. The study of video popularity churn and population are left as future work.

- Peer population is much larger than video population [3].

- All videos have the same size. The cloud storage stores only a small subset of the videos. Simulation usedheterogeneous videos length obtained from practical system [6].

- All peers have the same upload capacity. Peer's download capacity is much more than upload capacity in most of the networks. The download capacity constraint is not considered in this work.

- One downloading request is issued by each peer at any time. A multiple concurrent request model will be considered in future study.

- Peers are fully connected, forming a full mesh topology [3]–[5].

- The cloud server is able to replace any cached video with a new video instantly. That means the time for video replacement is ignored.

- The video size is large enough to be divided into an infinite number of small chunks[4], [5].

To simplify the scheduling strategy a homogeneous upload capacity is used. In this work, each peer shares their upload capacity to all other peers downloading the same video equally. The tradeoff between fairness and download performance in P2P file downloading system is discussed in paper [7]. Only homogeneous scenario is implemented as the focus is on download performance.The notations followed in this paper are given below

- N is the peer population. Since one peer only issues one request, the peer population is equal to request population. The peer population for a video j is denoted by Nj.

- M is the set of all videos. The total number of videos is M.

- K is the subset of videos cached by cloud storage. The number of cached videos is K. M – K is the set of videos not cached by cloud storage.

- Each video has a relative popularity , i.e., the probability that the video is the target of any request. By definition, = 1. =1

- Peer upload capacity is U. The upload capacity of the server (that sources a video content) is Uo.

- The (total) upload capacity of the cloud server(s) is H.

- Each video's file size is denoted by F.Multiple requests can be served by the cloud server simultaneously. A simple fair sharing scheduling strategy is followed i.e., the cloud server allocates its upload bandwidth evenly among all video requests being served.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

III.STATIC MODEL:

In analyzing with static model, let N denote the number of peers and downloading requests. The number of requests remains unchanged as the model is static. The cloud server can be thought of forming a logical start network, given the full mesh, and uplink limiting assumptions, the peers, server as shown in the fig. 1. The helpers included in the network are not interested in any movie , they just amplify the system capacity for the improvement in user experience. If the cloud server is given as helper, then Helper mode and Server mode are the two intuitive strategies to serve each downloading request.

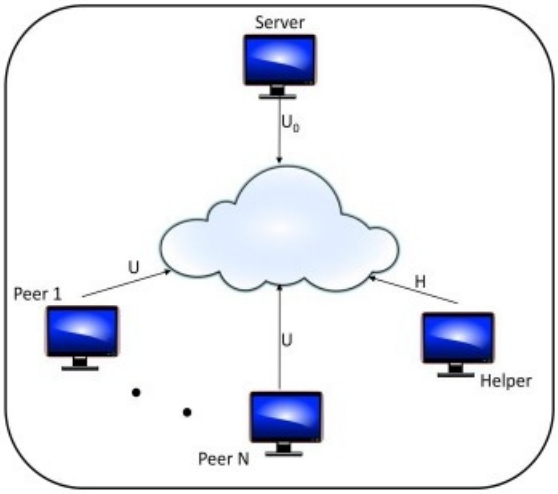

Fig.1: File Downloading with Helper

A single movie is considered for analyzing the static case. The performance objective of minimizing downloading time is equivalent to maximizing the throughput rate. The maximum throughput can also be referred to as the capacity of the P2P system, denoted as C, and it has been derived by previous studies. For example, the derivation of the capacity without helpers can be found in [4], [5], as = min 0 , + 0 . Capacity of the source (server) and total capacity are the two bottlenecks. By constructing a set of 1-hop and 2-hop spanning trees, [4] and [5] derive the maximum throughput and find a centralized scheduling strategy to achieve the maximum throughput. The cloud server needs to download video content from P2P or the source, if it adopts helper mode, which is a waste of bandwidth when compared with server mode. The cloud server should distribute particular chunks to all peers to maximize the throughput. Therefore the –1 efficiency to utilize cloud servers upload capacity is.

Fig.1: File Downloading with Helper

A single movie is considered for analyzing the static case. The performance objective of minimizing downloading time is equivalent to maximizing the throughput rate. The maximum throughput can also be referred to as the capacity of the P2P system, denoted as C, and it has been derived by previous studies. For example, the derivation of the capacity without helpers can be found in [4], [5], as = min 0 , + 0 . Capacity of the source (server) and total capacity are the two bottlenecks. By constructing a set of 1-hop and 2-hop spanning trees, [4] and [5] derive the maximum throughput and find a centralized scheduling strategy to achieve the maximum throughput. The cloud server needs to download video content from P2P or the source, if it adopts helper mode, which is a waste of bandwidth when compared with server mode. The cloud server should distribute particular chunks to all peers to maximize the throughput. Therefore the –1 efficiency to utilize cloud servers upload capacity is.

The maximum throughput in helper mode is $C = \min U_0, U + \frac{N-1}{N}U_0 + NH$

The maximum throughput in server mode is $C = \min U_0 + H, U + \frac{U_0 + H}{N}$

In static model, with the cost of additional storage to cache a particular video one can say that, the server mode is always better than the helper mode.

IV.DYNAMIC MODEL:

In dynamic model, the effectiveness of distributed scheduling using a parameter is modeled, in a fashion similar to the sharing efficiency parameter used in [3]. In this section, the discussion is extended to multi-video system. The arrival of peer downloading requests resembles a Poisson process. A peer leaves the system immediately, once the downloading gets completed. Uo is ignored to simplify the expressions, since it is a constant and it is insignificant. For exchanging video content between peers, a distributed chunk scheduling strategy is adopted. The base paper [2] analyzed the unlimited-cache case, helper mode and Server mode.In unlimited cache case, it is assumed that the cloud storage is able to store all videos i.e., K = M. In helper mode case, it is assumed that the cloud server serves all downloading request, whether it is cached or not. The cloud server will relay and amplify the video content from other peers, if it is not cached.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In server mode case, the request for video is blocked until the cloud storage gets updated for a video not cached. In this case, it is necessary to differentiate the peers as downloading and waiting peers.

V.IMPLEMENTATION OF ADAPTIVE ALGORITHM:

From the above static and dynamic analysis results, both server mode and helper mode has strengths and drawbacks. The P2P resource gets wasted in helper mode because the cloud server needs to keep download new content to help peers. While the server mode wastes the bandwidth resource of the blocked peers. The adaptive algorithm from [2] is furnished below. This algorithm determines the service mode for each movie. To determine the mode of each movie the cloud server adjusts the strategy periodically. This adjustment is done by running Automatic Mode Selection (AMS) Algorithm. The movies in helper mode are having higher priority to get included in the cloud storage. Other movies are considered in the decreasing order of peer population.

Algorithm : Automatic Mode Selection Algorithm:

For each movie j not in $\mathbf{K}$ do
  If the active movie is less than K then
    update cloud storage to add movie j by replacing any movie without request.
    $N' = N' + N_j$
  Else
    If $\frac{H}{N'} + \alpha_j N_j < N_j U$ then
      use helper mode for movie j
      $N' = N' + \alpha_j N_j$
    else
      keep blocking peers requesting for movie j
    end if
  end if
end for

Based on the analysis made above, additional bandwidth cost to download is the weakness of helper mode and the benefit is that more peers contribute their uploading capacity by switching their respective state from waiting to downloading. Therefore the adapted algorithm compares cost and benefit and it starts the helper mode if the benefit is larger than cost.

VI.RELATED WORK:

This work is inspired by the recent paper [2] the introduced the theoretical implementation of Adaptive Mode Selection Algorithm. To implement this algorithm an Object storage cloud is needed. A private object storage cloud is created using opensource in the organization. Based on the comparative studies of cloud frame works [8]-[12] OpenStack is selected. VMWare Vsphere is used to create hypervisor with bare metal virtualization. Then a storage cluster has been configured using six servers in the institutions lab and its networking infrastructure. All the nodes have same hardware configuration and are connected over a dedicated storage VLAN through Gigabit Ethernet links. OpenStack Swift is used to create storage environment [13].

It is a popular cloud storage system that provide object based access to data. Authentication can be provided in three ways [14]. TempAuth is selected for providing authentication. Disks of the storage nodes are formatted with an xfs file system. Each storage node is assigned to a separate zone and the replicas parameter has been set to 3. Then the cloud server is connected in the network. An application is developed with the implementation of Algorithm. Memcached tool is used for cache memory management. But the anlaysis slightly vary when compared with the theoretical calculations given in [2], because in practical systems, it is not possible to split the files into infinite number of chunks.

VII.CONCLUSIONS:

In this work, an object storage cloud with opensource software is implemented successfully. By following the theoretical model of analyzing the strategies for a cloud downloading system and it proposed algorithm, a system to download videos from cloud server is created. Two different diesign philosophies of the algorithm are helper mode and server mode. It is observed that helper mode wastes the bandwidth of the server but it leverages P2P capacity when the requests load is too high. It is also observed that the server mode is efficient for dealing with large video population relative to the cache size. This implementation helped a cloud downloading system to optimize its design.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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]Yipeng Zhou, Tom Z. J. Fu, Dah Ming Chiu, Yan Huang, —An Adaptive Cloud Downloading Service in IEEE Transactions on Multimedia, Vol.15. No.4, June 2013.

[3]D. Qiu and R. Srikant, —Modeling and performance analysis of bit torrent like peer to peer networks in Proc. ACM Sigcomm, 2004.

[4]J. Mundinger, R.Weber, and G.Weiss, —Optimal scheduling of peer-to-peer file dissemination, J. Scheduling, vol. 11, no. 2, pp. 105–120, Apr. 2008.

[5]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.

[6]QQ. [Online]. Available: http://xf.qq.com/

[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.

[8]Eucalyptus URL: http://www.eucalyptus.com/Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G.,Soman, S., Youseff, L. et al. (2009) The Eucalyptus Open-Source Cloud Computing System. In: 9th IEEE/ACM International Symposium on Cl uster Computing and the Grid .Shanghai, China 2009.

[9]B. Sotomayor, R.S. Montero, I.M. Llorente, I. Foster, Virtual infrastructure management in private and hybrid clouds IEEE Internet Comput. 13 (2009) 1422. doi:10.1109/MIC.2009.119. OpenNebula URL: http://opennebula.org/

[10]Nimbus. URL: http://www.nimbusproject.org/.

[11]OpenStack Swift URL: http:// http://www.openstack.org/

[12]OpenStack Swift Authentication System URL: http://docs.openstack.org/developer/swift/overview_auth.htmlR. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, —High-speed digital-to-RF converter, U.S. Patent 5 668 842, Sept. 16, 1997.