Research Journal of Applied Sciences, Engineering and Technology 7(24): 5283-5287, 2014
DOI:10.19026/rjaset.7.926
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: March 13, 2014 Accepted: April 25, 2014 Published: June 25, 2014

Research Article

Efficient Content Distribution in P2P and Cloud Service

E. Priya, A. Samydurai and T. Amitha
Department of CSE, Dhanalakshmi College of Engineering, Anna University, Chennai-601301, India

Abstract: Increasing the performance of cloud downloading service in a peer-peer network by efficient mode selection method. Using P2P to download video content often faces challenges like high bandwidth requirement, dealing with a huge number of peer requests, resulting in inefficient system performance. The latest premium service to have been developed is the cloud downloading. In this service, cloud storage is used to store the video content. The server is used to cache the user interested video content and also update the cache based on the peer request. Efficient Video content downloading using the P2P approach is accessible, but does not always give good concert. The cloud storage and server caches user-attracted content and updates the cache based on user downloading requirements. If a requested video is not in the cache, the request is detained in a waiting state until the cache is efficient. In this study, an alternative mode is proposed. The model contains the two modes named as server mode and helper mode. These two modes are integrated in cloud downloading service. Server mode is used for enhancing video content distribution in cloud storage. Helper mode is used to provide the user with the requested video content as soon as the peer request arrives. The two modes automatically switch between each other, based on the peer request. If the requested content is a popular video, helper mode provides the content. Whereas, the content is provided by the server mode if the requested content is unpopular. These two modes achieve good performance.

Keywords: cloud server, Cloud storage, file download, file upload, peer-to-peer, video content

INTRODUCTION

The principle of cloud download has helped in achieving high quality video content distribution by using cloud utilities. When the user sends video request to the cloud, the cloud downloads the video from the internet and stores it in cloud cache. The user retrieves the requested video with high data rate in any place at any time via the intra cloud data transfer system (named video cloud) (Huang *et al*., 2011). Content Delivery Networks (CDNs) overcome the limitations of the internet related to how the users recognize Quality of Service (QoS). In this study the CCDN problem of high cost and surrogate replacement are dealt by decomposing the problem to graph partitioning. This study aims at minimizing the average Shortest Path Betweeness Centrality (SPBC) and assigns each user a cloud, satisfying QoS requirements by assigning each end user at least one cloud site (Papagianni *et al*., 2013). The load management optimization involves two stages as layer replication and dereplication as proposed. The performance of the load management system can be compared on different aspects. Varying storage at the data sources compares the performance of the policies and has an impact in storage availability. Impact of replication bandwidth on performance occurs when the bandwidth available replication is varied (Venkatasubramanian and Ramanathan, 1997). To remedy the slow downloading rates for peers downloading unpopular videos, a cloud downloading service is deployed. There are two generic service modes for cloud servers namely server mode that focuses only on serving the content already cached at the cloud storage system and the second mode where the cloud server does not block any requests. we integrate these two modes into a single adaptive cloud downloading service using the adaptive Algorithm (AMS) (Zhou *et al*., 2013). Existing systems of cloud computing using VM addresses the issues on a SAN cluster by deduplication in a decentralized fashion at higher costs. So we propose Liquid, a distributed file system that has a client side breaking the VM images into small data blocks, referencing them by their fingerprints and using deduplication techniques and saving them into meta server. Whenever a VM image is needed a Liquid client downloads the fingerprints from the meta server, fetches data blocks from data servers exporting them from an integrated VM image layout to hypervisors (Zhao *et al*., 2013). Though many techniques have been devoted for provisioning the data fast, responding to large number of dynamic requests is not so successful. In this study we design VMThunder, a fast VM image provisioning tool for deploying a VM at a large scale that downloads only the on demand

Corresponding Author: E. Priya, Department of CSE, Dhanalakshmi College of Enginnering, Anna University, Chennai-601301, India

blocks during the startup phase. To handle a large number of requests at the same time, we adopt a P2P image streaming method that reuses the common data blocks cached across the nodes in the cluster (Zhang *et al*., 2014).

The objective of the proposed methodology is to increase the system performance improve the throughput of video issuing service in cloud peer-peer network. Cloud serve serves the video as soon as the request arrive. The helper mode is good at hold a high request rate, while the server mode is good at climbing with video population size. The mode can be changed due to peer request. If the requested video is popular video means helper mode provide the video. If the requested video is unpopular means server mode provide the video.

LITERATURE REVIEW

The latest premium services provided for video content downloading is called as cloud downloading. Consisting of two modes namely server mode and helper mode. Both these designs are useful for certain operating regimes. Here to propose an algorithm called Automatic Mode Selection (AMS) that switches between the modes automatically. Thus, it helps to optimize the design of cloud downloading services (Zhou *et al*., 2013). Yan *et al*. focused on using cloud download scheme for providing satisfactory content distribution service for unpopular video by improving the data transfer rate and by utilizing the cloud to guarantee the data health of video (Huang *et al*., 2011). Nalini and Srinivas proposed a predictive placement policy that determines the degree of replication based on priori predictions of injected subscriber requests for popular videos. This policy can be combined with an adaptive scheduling policy which determines the relative utility of resources in a video server for assignment of request to replicas. This combination performs best for more server configurations as load management procedure (Venkatasubramanian and Ramanathan, 1997). Peer-to-Peer overlay networks such as Bittorrent and Avalanche are highly implemented on large files for higher performance in an transaction from the server to as many end-users. The core concept is that the files are splitted into equal size in order to be downloaded by the users from the host. However the performance haven't reached the higher merits when compared to one system and the other new concept is that, analytical performance analysis based on the new uplink sharing version of broadcasting problem (Mundinger *et al*., 2008). Pure P2P architecture implement resource mediation and resource control mechanisms completely detached manner. Whereas hybrid P2P utilizes the core entities. All the peers who are accustomed from are served from many core index servers. The connection between the peers made through P2P proxy are by utilizing a cache depending on content popularity (Andersen *et al*., 2004). RCM is a trusted model that meets the security threats in P2P storage. Finding the best path for each neighbor is an issue in RCM for P2P storage. Each neighbor is usually involved with more than one resource path and these can be made use of. Binary tree which is an easy and commonly used tree can be employed in RCM (Lee and Lee, 2012). With embedding of many sensors, connectivity to the Internet and power being their advantage, smart phones have become an essential source of multimedia content and an important data sharing tool. In this demo, we intend to deal with video streaming in smart phones. To present a novel approach called StreamSmart which forwards the captured content to its own cloud clone that forwards to other such clones, making StreamSmart highly fault tolerant, scalable and responsive (Gaeta *et al*., 2013). This study deals with improvement of file access performance. This study describes HDFS-based Distributed Cache System. The cache services which are a part of the cache system are designed with three access layers. The shared memory caches the files loading from HDFS which can then be accessed by a client library (Zhang *et al*., 2012). This study proposes a self adaptive load balancing algorithm that meets the problem of network traffic in P2P networks. In our algorithm, the nodes create binary tree and back-up node tables. Demonstrations prove that our algorithm can reduce load and balance high speed network nodes (Xiong *et al*., 2012). Detection of colluders is a major task to preserve data from unauthorized peers. Our study provides two schemes that detect as many peers as possible with avoidance of misdetections. The first scheme collects reports from all the participants while the second one applies a technique to improve the quality of colluder. Experimental results involving 10% colluders show that our algorithm detects all colluders accurately (Abdullah and Fujita, 2012).

With multimedia growing exponentially in today's internet, content delivery networks play a key roles. The current cloud delivery systems have pre-allocated storage that is usually over provisioned. In this study, experimental results demonstrate that the proposed system deals dynamically with provision storage to facilitate efficient sharing of multimedia content (Andersen *et al*., 2004). In this study, we propose a cloud-assisted power efficient mobile P2P media streaming architecture. This addresses the weakness of wireless access technologies. Clouds are responsible for storage and computation of the stored data. We model interactions among mobile devices as a coalition games and then deal with the optimal chunk retrieval scheduling (Jin and Kwok, 2010).

In this study, we propose a hierarchical framework and evaluate it towards an efficient and scalable solution for content distribution. Novel proposals for the replica placement problem within the cloud are proposed, taking the limitations of the physical substrate into consideration. The performance of the proposed framework is assessed through modeling and simulation with definition of appropriate metrics (Blair *et al*., 2012).

In order to deal with the challenges in resource provisioning and replica placements, this study proposes a set of algorithms. This solves the problems resource provisioning and caching. We propose DPC (Differential Provisioning and Caching) algorithm that aims to rent cloud resources to build CDNs and cache the contents to reduce the total rental cost. We also propose the CRB (Caching and Request Balancing) to adjust the placements and contents of route map, dynamically (Hu *et al*., 2013). A successful alternative to the conventional content delivery is the Cloud oriented Content Delivery Networks (CCDN). In this study, an efficient solution is provided for distributing content over multi-provider inter network environment. Novel replica placement algorithms are used to overcome replica placement problem for the virtualized environment (Papagianni *et al*., 2013). A traditional system using virtual machine cannot satisfy the increased demand for large scale VM hosting. So we propose Liquid, a scalable de-duplication file system designed particularly for large scale VM deployment leading to increased IO performance by caching frequently accessed data (Zhao *et al*., 2013). A way to respond to a large number of user requests using virtual machines is a challenge. We address this challenge with VMThunder, a new VM tool and speed up VM image streaming by integrating Peer-to-Peer (P2P) streaming techniques (Zhang *et al*., 2014).

MATERIALS AND METHODS

Figure 1 shows that the server uploads the videos to the cloud storage. The figure shows n number of peers (I) sending requests to download the videos. There are two generic service modes for cloud servers. In the first mode called the server mode (S_m), the cloud server is primarily focused on serving the content already cached at the cloud storage system. The cache consists of frequently requested data. Requests for content not in the caches are blocked until such content becomes cached. The cloud storage system updates its cache periodically to replace content without requests by content with requests awaiting. An alternative mode is the helper mode (H_m), in which the cloud server does not block any requests. For videos that are not cached, the cloud servers simply relay chunks from some peers to other peers, acting as a helper peer. One contribution of our study is to compare these two modes analytically.

Notation:

$$Um^{o} = \sum_{i=1}^{n} si$$

U_m = Upload capacity
S_i = Storage capacity
I = Number of peers
H_m = Helper mode
S_m = Server mode

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 a single adaptive cloud downloading service.

Figure 2 shows that the file uploaded by the server in cloud storage. The Server must register the user name and password for uploading the video. The uploaded video is stored in cloud storage. Next th peer requests are arrives to cloud storage. Cloud storage analyses the request whether it is a popular video or an unpopular one. If the peer requested video is popular then the helper mode will provide the content from cache as soon as it arrives. If the requested video content is unpopular, then the server mode searches for the video stored in cache then provides the video to the peer.

Efficient mode selection method: The server uploads the video content with the help of its username that is to be registered in cloud storage. A large number of Peer

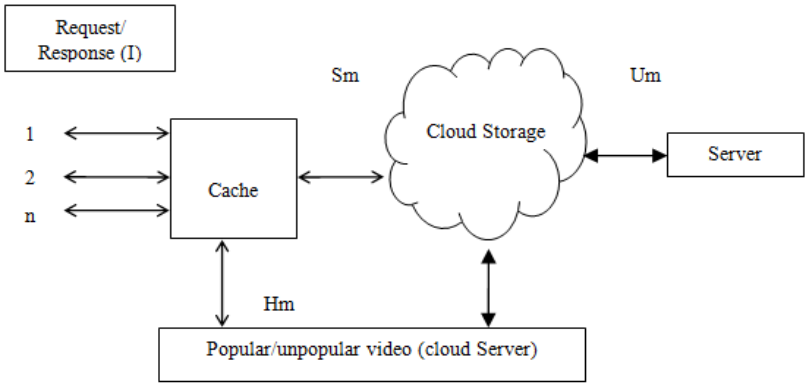

Fig. 1: Architecture diagram

Fig. 2: Cloud storage

requests arrive to download the video content. The cloud storage analyses the request to find if it is a popular video content or unpopular video content. Automatically the mode can be changed whether it is server mode or helper mode. Server mode provides the content that is unpopular. Helper mode provides the popular video content. The downloading time will hence be reduced and also this also effects in a good performance.

RESULTS AND DISCUSSION

In cloud downloading service, the user requested video content is downloaded. The CDN becomes the bottleneck for server capacity when there is a large number of requests for video downloading. The peers requesting unpopular videos grieve low downloading rate. The two modes helper mode and server mode are switch between each other based on the peer requested content. These are the two universal modes for cloud server. The cloud server serving the content already cached in the storage system works in the helper mode. If the content is not in the cache then the server blocks the peer requests until the server caches the requested content in storage. The helper mode doesn't block any request because it provides the video content that the peer requested, as soon as it arrives. More peers can fund their upload capacity by switching their state from waiting to download.

REFERENCES

Abdullah, E. and S. Fujita, 2012. Colluder detection in commercial P2P CDNs using reputation information. Proceeding of the 7th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), pp: 183-190.

Andersen F.U., H.D. Meer, I. Dedinski, C. Kappler, A. Mader, J.Oberender and K. Tutschku, 2004. An architecture concept for mobile p2p file sharing services. Proceeding of the Workshop Informatik-Algorithms and Protocols for Efficient Peer-to-Peer Applications, pp: 1-5.

Blair, A., P. Gerard, M. Philip and S. Bryan, 2012. Realtime utilisation of cloud storage. Proceeding of the 1st International Conference on Cloud Networking (CLOUDNET), pp: 188- 190.

Gaeta, A., S. Kosta, J. Stefa and A. Mei, 2013. StreamSmart: P2P video streaming for smartphones through the cloud. Proceeding of the 10th Annual IEEE Communications Society Conference Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 pp: 233-235.

Hu, M., J. Luo, W. Yang and B. Veeravalli, 2013. Practical resource provisioning and caching with dynamic resilience for cloud-based content distribution networks. IEEE T. Parall. Distr., 99: 1.

Huang, Y., Z. Li, G. Liu and Y. Dai, 2011. Cloud download: Using cloud utilities to achieve high-quality content distribution for unpopular videos. Proceedings of the 19th ACM International Conference on Multimedia, pp: 213-22.

Jin, X. and Y.K. Kwok, 2010. Cloud assisted P2P media streaming for bandwidth constrained mobile subscribers. Proceeding of the 16th International Conference on Parallel and Distributed Systems (ICPADS), pp: 800-805.

Lee, S. and W. Lee, 2012. Tree problems of the resource chain trust model for P2P storage cloud. Proceeding of the International Conference on ICT Convergence (ICTC, 2012), pp: 331-332.

Mundinger, J., R. Weber and G. Weiss, 2008. Optimal scheduling of peer-to-peer file dissemination. J. Sched., 11(2): 105-120.

Papagianni, C., A. Leivadeas and S. Papavassiliou, 2013. A cloud-oriented content delivery network paradigm: Modeling and assessment. IEEE T. Depend. Secure, 10(5): 287-300.

Venkatasubramanian, N. and S. Ramanathan, 1997. Load management in distributed video servers. Proceeding of the 17th International Conference on Distributed Computing Systems, pp: 528-535.

Xiong, N., K. Xu, L. Chen and L.T. Yang, 2012. An effective self-adaptive load balancing algorithm for peer-to-peer networks. Proceeding of the IEEE 26th International Parallel and Distributed Processing Symposium Workshops (IPDPSW, 2012), pp: 1425-1432.

Zhang, J., G. Wu, X. Hu and X. Wu, 2012. A distributed cache for hadoop distributed file system in real-time cloud services. Proceeding of the 13th International Conference on Grid Computing, pp: 12-21.

Zhang, Z., Z. Li, K. Wu, D. Li, H. Li, Y. Peng and X. Lu, 2014. VMThunder: Fast provisioning of large-scale virtual machine clusters. IEEE T. Parall. Distr., 1: 1.

Zhao, X., Y. Zhang, Y. Wu, K. Chen, J. Jiang and K. Li, 2013. Liquid: A scalable deduplication file system for virtual machine liquid. Proceeding of the Parallel and Distributed Systems, pp: 1.

Zhou, Y., T. Fu, D.M. Chiu and Y. Huang, 2013. An adaptive cloud downloading service. IEEE T. Multimedia, 15(4): 802-810.