

CloudTCast: A Video Distribution Solution Based on Cloud and Tree-based ALM

Weidong Gu^{1, 2, a}, Lu Wang^{2, b}, Bin Gong¹, Huiling Shi²

¹Department of Computer Science and Technology, Shandong University, Jinan 250101, China

²Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center(National Supercomputer Center in Jinan) , Jinan, 250101, China

^aemail: guwd@sdas.org, ^bemail: wanglu@sdas.org

Keywords: Cloud Computing; Application Layer Multicast; Multicast Group; Cloud Virtual Machine

Abstract. This paper studies a novel data distribution solution, named CloudTCast, for video multicast group. CloudTCast combines the advantages of cloud and tree-based application layer multicast (ALM). On the one hand, it uses the tree-based ALM to break the cloud's bandwidth limitation in delivering the video to large-scale concurrent users. On the other hand, it leverages cloud to effectively improve the performance of tree-based ALM. CloudTCast can make elastic extension in terms of the size of specific video multicast group.

Introduction

Cloud computing is a collection of applications, hardware and system software designed to deliver services to end users over the Internet [1]. It offers a wide range of services, including SaaS (Software as a Service), PaaS (Platform as a Service) and IaaS (Infrastructure as a Service). Cloud computing has the advantages of high reliability and scalability, on-demand service, low maintenance costs, and virtualization. Due to these advantages, it is considered as the next-generation computing paradigm, and inspires enormous effort to integrate the existing technologies with this computing model.

Migrating the video application with large-scale concurrent users (e.g., live streaming and video on demand) to the cloud is an interesting work because this application is very popular on the current Internet. Although cloud computing has many obvious advantages, it is difficult even impossible for the cloud to directly provide a large-scale video distribution because the limited bandwidth of cloud's outbound links cannot satisfy the high bandwidth requirement of this application [2].

Multicast is widely considered as the ideal way to distribute data from a source to many receivers. Most multicast protocols use tree as the data delivery structure because the delivery tree has good forwarding effectiveness. Multicast solutions can be divided into two main categories, i.e., IP multicast and ALM. ALM implements the multicast functionality at common users instead of the routers, which addresses the deployment problem of IP multicast. Bonald et al. [3] studied several tree-based ALM protocols and proved that some of them can achieve near optimal rate in static streaming systems. However, the tree-based ALM is essentially unreliable because any user can randomly leave the multicast group. More specially, when an end user leaves the group, its downstream nodes suffer some data loss or high packet transmission delay during the tree rebuilding procedure. Additionally, most existing ALM solutions build the delivery trees by the distributed way, which is difficult to optimize the trees from an overall point of view.

In this paper, we study a novel video multicast solution, named CloudTCast, based on the combination of cloud and tree-based ALM. The combination is two-fold. On the one hand, we use the tree-based ALM to break the cloud's bandwidth limitation in the data distribution for large-scale video multicast group. On the other hand, we leverage cloud to alleviate the unreliability problem of tree-based ALM. CloudTCast uses the cloud to store and publish the videos, and uses cloud-assisted tree-based ALM to effectively distribute data.

Related Work

Unlike cloud computing, peer to peer way can complete large-scale data distribution. So far there exist a lot of P2P streaming protocols, which generally fall into two coexisting strategic categories—mesh-pull P2P and tree-push P2P (i.e., tree-based ALM). The tree-based ALM solutions (e.g., [4], [5], [6], [7], [8]) organize participating peers into one or more multicast trees, and disseminate the data along these trees. As mentioned above, the tree-based ALM can obtain excellent forwarding effectiveness. In the mesh-pull P2P streaming protocols (e.g., [9], [10],[11]), the data is presented as a series of segments, each representing a short duration of playback. Every peer maintains a list of neighboring peers, and periodically exchanges segment availability information of data buffers (often referred to as buffer maps) with its neighboring peers. The mesh-pull P2P is more resilient to the peer dynamics than the tree-based ALM. However, the mesh-pull P2P has low forwarding effectiveness [3].

Recently the data distribution based on the combination of cloud computing and mesh-pull P2P has gradually become an interest in the research community (e.g., [12], [13], [14], [15], [16]). [12] presents a use case of the cloud infrastructure, by introducing an architecture for P2P multimedia streaming in both CS and P2P style. [12] offers QoS API functions implemented in a web service of the cloud streaming service provider, which enables users to decide about the contract type to establish with the service provider or go for P2P streaming as a possible solution. Islam et al. proposed a dichotomy between the user and cloud-based services by using an edge-based computation, coordination and storage facility [15]. This extension is useful for the support of multimedia applications, but it seems to be difficult to widely deploy in the real network. [16] presents a P2P resource search mechanism which adopts P2P networking technologies to orchestrate all the computing resources.

CloudTCast Architecture

In CloudTCast, there are two types of users—cloud user and end user. The cloud user here denotes the video provider who uses CloudTCast to distribute the video to one or multiple groups of users. The end user represents the receiver of the video. For each cloud user, CloudTCast uses a cloud virtual machine (called core VM) as the main infrastructure and n ($n \geq 0$) virtual machines (called relay VMs) to extend the bandwidth of the core VM. These VMs can be customized in terms of their roles, the volume of videos and the scale of users. In general, the core VM has high computing capability and good outgoing bandwidth, the relay VM has common computing capability and good outgoing bandwidth. The memories of these VMs are customized or resized in terms of the volume of videos because these VMs cache received video data for a designed time (e.g., 3 mins). Most existing IaaS services can satisfy the above requirements. The cloud that offers the core VM is said to be the core cloud. Relay VMs can be obtained from the core cloud or other clouds. CloudTCast is designed to serve multiple concurrent video multicast groups from the same cloud user.

Fig. 1 shows main functionalities of CloudTCast which are described below

Video upload and storage. The cloud user uploads the video to the core VM. If the user wants to instantly distribute the video streaming, then the video data is cached in video data buffer of the core VM and is stored if needed. If the user uploads the video in advance, the video is directly saved in the storage. During the multicast session, the video data is cached in the memory for a relatively long time so that the data distribution module can quickly obtain the required data. Each cloud user is assigned a unique account, which is saved in the storage.

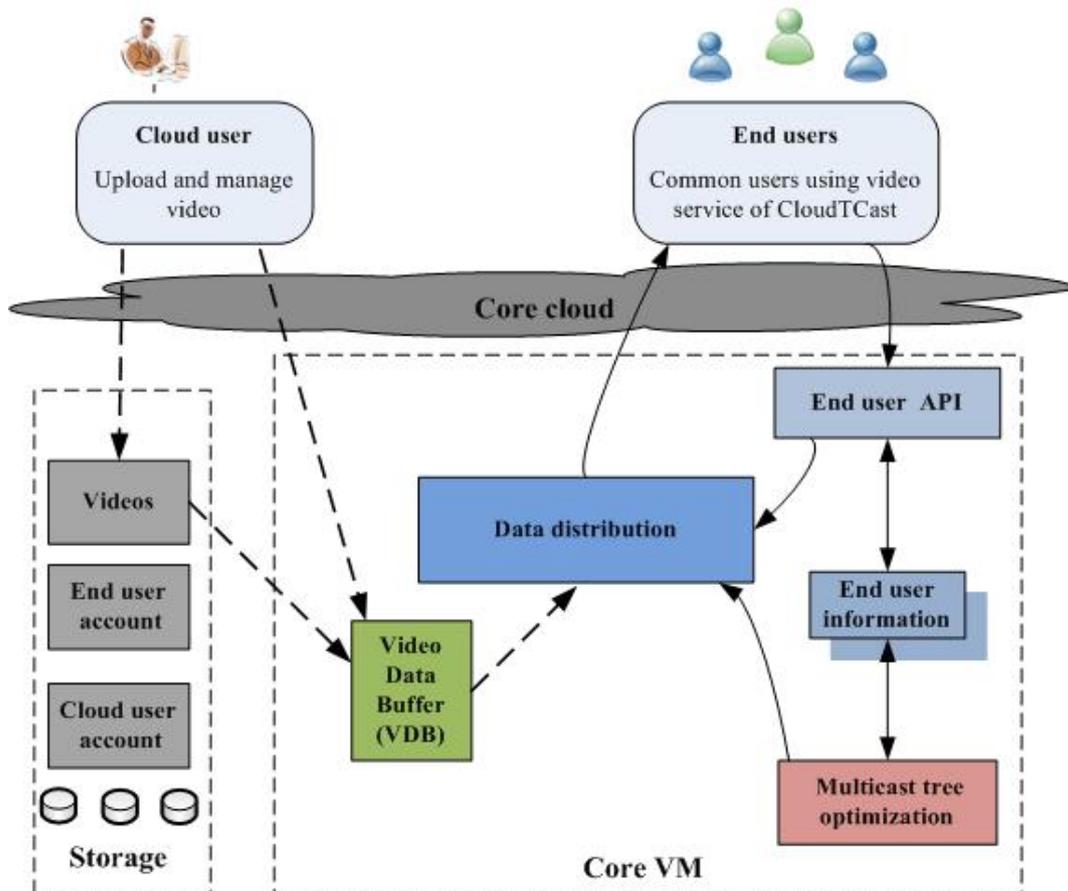


Fig. 1. Components and Functionalities of the core VM

Data distribution. CloudTCast creates a core VM and multiple relay VMs for each cloud user. CloudTCast uses these VMs and end users to jointly perform the data distribution, as Fig. 2 shows. The fanout (i.e., the maximum number of children that a node is willing to accommodate in the multicast tree) of the virtual machine for a group is limited by the allocated bandwidth for this group. In this paper, we use $f(e)$ to denote the fanout of the node e , and use $f'(e)$ to mean the remaining fanout of e . Note that $f'(e) = f(e) - m$, where m denotes the number of existing children of e . The relay VM obtains the video data from the core VM.

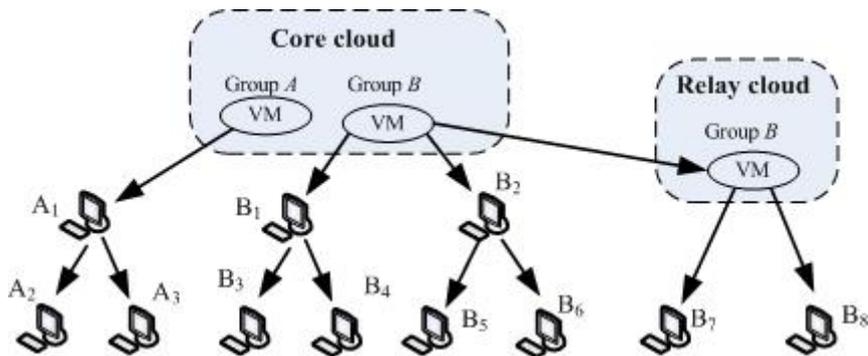


Fig. 2. CloudTCast multicast tree

End user API, information and account. The end user accesses the core VM through the end user API. CloudTCast saves the information related to the multicast session and end users, including the times when the user joins and leaves the multicast session, the distance (using latency as the metrics) between two adjacent nodes, the fanout of the end user, etc. CloudTCast supports two types of end users, i.e., anonymous end user and registered end user. The registered end user can accumulate its contribution which can be used to obtain more services. The contribution is related to user's stability. We use the contribution to encourage the end user to keep stability, to alleviate the unreliability of ALM as an extra way. The end user account module computes the

contribution of each registered end user.

Multicast tree optimization. The multicast tree optimization module uses the end user information saved in the core VM to produce the logical tree structure corresponding to the CloudTCast multicast tree, and further optimize the logical tree structure. If the optimization scheme can obtain desirable gains, then CloudTCast adjusts the multicast tree according the scheme. In this paper, we do not further discuss the multicast tree optimization.

CloudTCast Multicast Tree Construction

In CloudTCast, the multicast tree is gradually built as end users join the video multicast session. CloudTCast uses the revised version of any existing ALM tree building algorithm. In the revised version, an end user e whose fanout is larger than 0 can become the child of another end user p by replacing one 0-fanout child of p . If an end user replaces a 0-fanout node, it is responsible to accept the node as its child node. Fig. 3 describes the CloudTCast multicast tree building algorithm. Note that the procedure ignores the joining procedure of relay VM nodes. From the algorithm, we can notice that the end users first attempt to obtain the multicast data from the VM nodes. Since the VM nodes are stable, the above feature can effectively improve the multicast performance.

Algorithm 1:The newcomer m joins the multicast session S

1. Initialize a null stack K ;
 2. m obtains the relay VMs for the core VM for the multicast session S ;
 - \setminus Assume that V is the set of these VMs
 3. m sends distance measurement messages to the VMs in V , to obtain the distances from these VMs to it;
 4. The identifications of the VMs in V are inserted into the stack K , in the descending order of VM-to- m distances;
 5. $min \leftarrow MAXD$; // $MAXD$ denotes a constant that is larger than the maximum network distance
 6. $cvm \leftarrow null$; $conti \leftarrow true$;
 7. While K is not null and $conti == true$ do
 8. Pop the top element, denoted by v , of K ;
 9. If $f(v) > 0$ then
 10. m becomes a child of v ;
 11. $conti \leftarrow false$;
 12. ElseIf the distance from v to m is less than min
 13. $min \leftarrow$ the distance from v to m ;
 14. $cvm \leftarrow null$;
 15. EndIf
 16. EndWhile
 17. If $conti == true$, then m joins the subtree rooted at cvm in terms of existing ALM algorithm.
-

Fig. 3. ALM tree building algorithm

Experimental Results

In this section, we evaluate CloudTCast using NS2 [17] as the simulator. We use the GT-ITM Generator [18] to create a 5,000-node transit-stub graph as our underlying network topology. The

VM nodes are fixed at different stub-domain nodes, and 1000 end users are located randomly among the other stub-domain nodes. We investigate CloudTCast systems using HCcast [5] as the ALM tree building algorithm. In our experiments, the fanout of any end user is valued by a random integer from 2 to 5.

To evaluate the performance of CloudTCast, we introduce the delay optimization ratio, which indicates the ratio of the delay of HCcast to the delay of CloudTCast. Fig. 4 and 5 present the mean and maximum delay optimization ratio in 10 multicast groups. From the figures, we can see that CloudTCast can effectively reduce the delay.

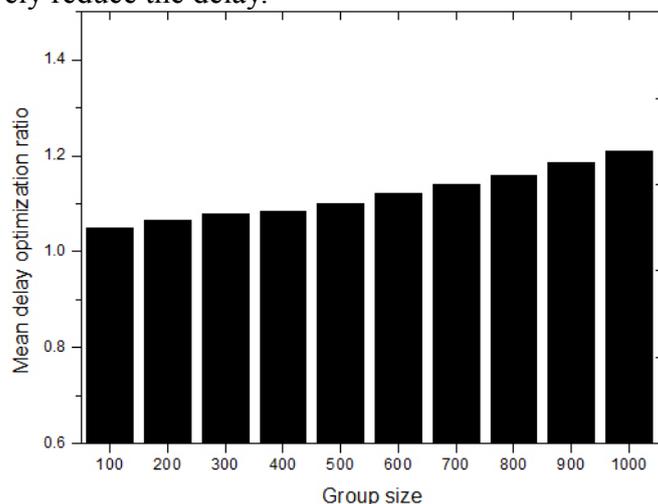


Fig. 4. Mean delay optimization ratio in 10 groups

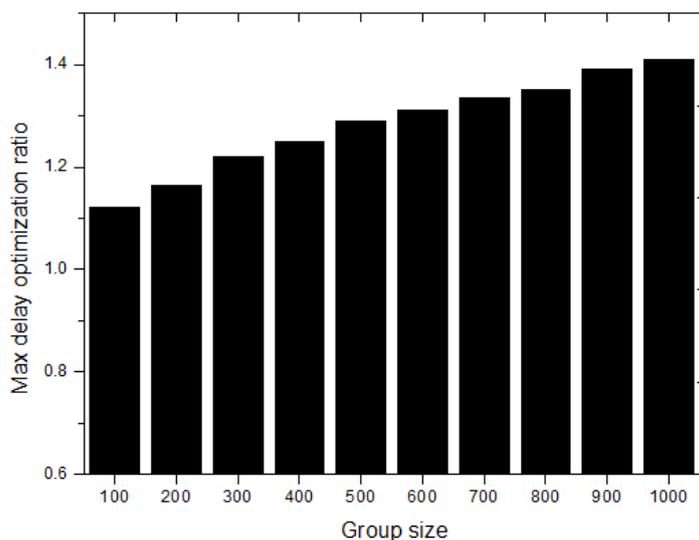


Fig.5. Maximum delay optimization ratio in 10 groups

Conclusion

This paper proposed a novel data distribution solution, CloudTCast, for the video multicast group applications. CloudTCast takes advantages of the tree-based ALM and cloud. Specially, it uses the tree-based ALM to assist the cloud in delivering the video to multicast groups, and uses cloud to effectively optimize the tree-based ALM.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant No. 61272433 and 61472230, and the Shandong provincial Natural Science Foundation of China under Grant No. ZR2013FM010 and ZR2014YL042.

References

- [1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia. Above the clouds: a berkeley view of cloud computing. Technical Report No. UCB/EECS-2009-28, 2009.
- [2] N. Leavitt. Is cloud computing really ready for prime time? [J]. *Computer*, 2008, 42(1) 15-20.
- [3] T. Bonald, L. Massoulie, F. Mathieu, D. Perino, A. Twigg. Epidemic live streaming: Optimal performance trade-offs [C]. *Proc. of ACM SIGMETRICS'08*, 2008.
- [4] L. Mathy, R. Canonico, D. Hutchison. An overlay tree building control protocol [C]. *Proc. of NGC'01*, 2001.
- [5] X. Zhang, X. Li, W. Luo, B. Yan. An application layer multicast approach based on topology-aware clustering [J]. *Computer Communications*, 2009, 32(6) 1095-1103.
- [6] B. Zhang, S. Jamin, L. Zhang. Host multicast: A framework for delivering multicast to end users [C]. *Proc. of IEEE INFOCOM'02*, 2002.
- [7] S. Banerjee, B. Bhattacharjee, C. Kommareddy. Scalable application layer multicast [C]. *Proc. of ACM SIGCOMM'02*, 2002.
- [8] Y. Y. Lin, J. Y. B. Lee. Path selection in streaming video over multi-overlay application layer multicast [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2010, 20(7) 1018-1031.
- [9] D. Frey, R. Guerraoui, A. Kermarrec, M. Monod. Boosting gossip for live streaming [C]. *Proc. of IEEE P2P'10*, 2010.
- [10] X. Zhang, J. Liu, B. Li, T. shing, P. Yum. Coolstreaming/donet: A data-driven overlay network for peer-to-peer live media streaming [C]. *Proc. of IEEE INFOCOM'05*, 2005.
- [11] N. Magharei, R. Rejaie. Prime: Peer-to-peer receiverdriven mesh-based streaming [C]. *Proc. of IEEE INFOCOM'07*, 2007.
- [12] I. Trajkovska, J. Salvach ´ ua, A. M. Velasco, C. Alonso. A novel p2p and cloud computing hybrid architecture for multimedia streaming with qos cost functions [C]. *Proc. of ACM MM'10*, 2010.
- [13] A. H. Payberah, H. Kavalionak, V. Kumaresan, A. Montesor, S. Haridi. Clive: Cloud-assisted p2p live streaming [C]. *Proc. of IEEE P2P'12*, 2012.
- [14] X. Jin, Y.-K. Kwok. Assisted p2p media streaming for bandwidth constrained mobile subscribers [C]. *Proc. of 16th International Conference on Parallel and Distributed Systems*, 2010.
- [15] S. Islam, J.-C. Gr´egoire. Giving users an edge: A flexible cloud model and its application for multimedia [J]. *Future Generation Computer Systems*, 2012, 28 823-832.
- [16] Y.-F. Yu, K.-C. Lai. A semi-structured overlay for multiattribute range queries in cloud computing [C]. *Proc. of the 13th IEEE International Conference on Computational Science and Engineering*, 2010.
- [17] The network simulator—ns-2. [Online]. Available:<http://www.isi.edu/nsnam/ns/>
- [18] E. W. Zegura, K. Calvert, S. Bhattacharjee. How to model an internetwork [C]. *Proc. of INFOCOM'96*, 1996.