A Kind of proxy caching program based on doubly linked list in VOD system

Jianzhong HOU, Qiaolin CHAI*,
College of Computer Science and Technology, Shandong University, 1500 Shunhua Road
Jinan, Shandong P. R. China, 250100
Houjz2010@foxmail.com

Abstract. The VOD (Video-on-Demand) service has come to a reality because of the development of computer network technology and digital technology being more and more mature. But it demands high bandwidth requirement, high quantity data volume and strong real-time, these make the network bandwidth become the bottleneck of the VOD's development. So, an agent-based distributed VOD model appeared. We present a useful proxy caching scheme based on doubly linked list according to this model, it could settle the problem effectively which server load and network bandwidth base the current network equipment.

Keywords: VOD，Proxy caching，Doubly linked list

1 Introduction

With the development of Internet technology, more and more user began to abandon the traditional passive reception and viewing of multimedia, which made video on demand services became a reality. VOD is a kind of streaming media player application system that users can get any type of multimedia data to the client end to be broadcast from the server at any time [1].In view of the existing network infrastructure, server load and network bandwidth become the main limited factors in the widely using of video streaming. Set proxy, between the server and the user, is considered as a highly effective solution to reduce the bandwidth consumption. This system is called distributed VOD system [2]. Somebody proposed a variety of cache scenarios in accordance with different situations [2-7]. In these cache scenarios, the typical scenario is to proxy cache the initial part of the program (program head, Prefix) [2]. However, how to determine the size of the cache to effectively maximize the proxy role has become a very real problem.

1.1 Distributed VOD system model

The system model includes the server, caching proxy, and user clusters.

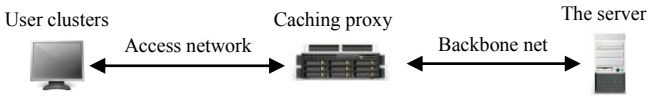

Figure 1 Distributed VOD model

Server and caching proxy is connected through backbone net. Caching proxy and user clusters are connected through access network.

Users get video on demand service through proxy from the server. Assume that the user always wanted to watch the video from the beginning, after the proxy receives the user's request, if the video head was cached on the local proxy, then the proxy transfers it directly to the user; if the program was not fully cached in local proxy, then proxy requests the program end from the server, and then transfers to the user [7].

2 Program and related parameters

In response to these models and a variety of network conditions people have proposed a variety of programs, existing programs as follows:

2.1 Analysis of existing programs

Existing mainstream programs are as the followings

2.1.1 Batching program and patching program

Batching program is initially proposed in response to server-based -demand system. It is based on the combined flow ideas, using the same media streaming to provide on-demand services for multiple users who request the same program. Its implementation is based on multicast technology. After the introduction of cache in the VOD system, batching scheme has been applied to proxy-based-demand systems and the use of proxy cache prefix program to provide timely services without delay.

Patching program is also based on the multicast merged flows idea. Server time generate entire length of the full broadcast program stream SR (called Regular Stream) for r_0 user request arrived at t_0. At t_1, r_1 user request arrives, and the server generates program fragment stream SP_1 (called Patching streams) which contains content [0, t_1-t_0]. User 1 receives these two program streams simultaneously, and immediately plays SP_1 stream data, and caches the SR stream data to disk. When finished playing SP_1 flow, the user 1 will play the SR stream data cached in the disk, and then the SR date that is receiving will continue to join the SR stream data disk buffer. Adoption of such technology, the program only requires a complete data stream and a very short clip streaming, you can satisfy both the user request r_0 and r_1. In t_2, r_2 user request arrives, similarly, the server program again generate a fragment containing [0, t_2-t_0] Patching stream of SP_2.The twice user finishes playing the SP_2 stream and then merge into the Regular stream SR generated for r_0.

Advantages: Ease the video server I / O bandwidth and network bandwidth limited by the using of multicast technology.

Disadvantages: does not consider on-demand strength. That is to say, all the programs stored in the proxy at the same length.

2.1.2 An optimal proxy cache allocation

Optimal Proxy Cache Allocation (referred to as the OPCA) is based on the global searching of excellent method to achieve the optimal program, which the proxy cache space is divided into a number of data units, and which distributes units one by one, until all data unit is distributed off, and the distribution of each unit are optimally allocated to a certain program.

Advantages: allocated the best local proxy cache space of each program to achieve the globally optimal allocation.

Disadvantages: initialization requires a lot of calculations .Each new program will also add a lot of calculations, especially when there are more programs. These two points lead to low proxy performance.

2.2 Programs based on double-linked list and related parameters

Because these types of programs still have some shortcomings, for the network bandwidth bottleneck phenomenon, this paper proposes a proxy cache solution based on doubly linked list.

2.2.1 Doubly Linked List

Also known as Doubly Linked List, it's a kind of list. It has two pointers in each data node that point to the direct successor and direct predecessor. Next node in each node stores successor address, besides, each node adds a point which named prior which points to its direct predecessor. The doubly linked list is determined only by the head pointer.

2.2.2Program and ideas

The proxy storage space is divided into three parts: one is mainly used to maintain the normal operation of the system; another is used to cache the information requested from the server; another is used to store video information. Store some of the settings in the video space and maintain two overall control parameters: One is the total storage space S, the other is the free space L. Create several (refered as N) doubly linked list in the space which stores video information. The value of N can be selected according to the actual performance of the proxy (under normal circumstances we take 5-10). According to the video on demand strength, the video information is distributed to different doubly linked list. And then assign different size video space for each video information based on the function F (V_i, L_i, R_i).

2.2.3 Function and relevant parameters

Function F (V_i, L_i, R_i) = (V_i / L_i) * R_i: V_i stands for the video playback speed; L_i stands for the length of the video information; R_i stands for the video information on demand strength.

The first two parameters can be obtained from the video information itself. For R_i, set up a counter for each video information. Each time a user requests a video on demand, the video of the counter plus one: the counter value is considered as the strengh of the video information on demand. That is the value of R_i.

In order to increase the flexibility of the function F (V_i, Li, R_i) to make it possible to adapt to a variety of network environments, we can take some improvements.

Function F '(V_i, L_i, R_i) = (p*V_i / L_i) * (q*R_i) ,in which p, q are constants, p and q can take their value flexibility based on the actual bandwidth, the proxy performance and the storage space.

2.2.4 Program implementation

Periodically check the counter value, record and then clear and after that, start count again. Then according to the recorded value of the counter, adjust the video to a different queue and calls the function F (V_i, L_i, R_i) to recalculate the space distributed for the video. If the new allocated space is less than the original space, then start to drop from the end of the video until it reaches the size of the new space, and the remaining space is for recycling. If the new allocated space is more than the original space, then request the end of the video information from the server, until fill the remaining space. When the re-allocated space is greater than the remaining space L, find the video information which has the smallest counter value from the minimum video-on-demand strength double-linked list, and then remove it. Recycle it space to the remaining space, and re-allocate it. If the re-allocated space is still larger than the remaining space L, then repeat the process of recycling space.

The time to check the value of this counter can't be set too short nor too long, if not it would cause the queue jitter or we will not achieve the proper adjustment.

At the proxy initialization, based on the actual situation, we can design video information on demand strength artificially, and allowed it to be added into the specified double-linked list, and then according to its following on-demand strength, dynamically adjust it periodically according to the program.

2.2.5 Program benefits

Proxy-based distributed VOD cache model which adapt double-lists structure has the following advantages:

1) The management of the programs according to the on-demand strength meets the actual application;

2) Reduce the user's response time. When a user requests a video, you can search in several doubly linked lists simultaneously. It reduces the time spent to search the targeted video, thereby reduce user's response time;

3) The calculate amount is small, and it won't affect the proxy performance. Whether adding a new program or in the proxy cache initialization, simply compare the value of R_i to determine the list, then call the function F (V_i, L_i, R_i) to compute the space. This calculation will be small and will not change with increasing number of larger programs;

4) The program is simple, easy to implement.

2.3 Applications and developments

Above we only consider the case of only one server. However, in practical applications, since there are a large number of users in user clusters, to improve the video service loss rate [9], we need multiple servers to work together. In order to improve server efficiency, we introduced the load balancer.

All servers are connected with the load balancer, and send their load information in real-time to load balancer. After the load balancer receives a request for information send by the proxy, assign proper sever to carry out the appropriate request processing according to the load of each server.

3 Conclusions

VOD system has a huge potential market and application prospect in the long run. It represents the application and development trends of future all-function network and digital interactive information. The server load and main network bandwidth bottleneck problem have been effectively mitigated, if adopting double-linked list-based proxy caching scheme.

References

1. Little T, Venkatesh D.: Prospects for interactive video-on-demand. J. IEEE Multi media，Vol1(3) ，1994，pp.14 -24
2. Sen S，Rexford J，Towsley D.: Proxy prefix caching for multimedia streams. A. IEEEI NFOCOM［C］. New York: IEEE Computer Society, 1999.1310 -1319.
3. Guo Y，Subhabrata S，Towsley D.: Prefix caching assisted periodic broadcast: framework and techniques to support streaming for popular videos. R. Tech Rep, UM- CS-2001-022. Amherst: University of Massachusetts Amherst, 2001.
4. Wang B，Sen S，Adler M，et al.: Proxy- based distribution of streaming video over unicast / multicast connections. R. Tech Rep, 01-05. Amherst: University of Massachusetts Amherst, 2001.
5. Verscheure O，Venkatramani C，Frossard P，et al.: Joint server scheduling and proxy caching for video delivery［EB / OL］. Http:\\www.citeseer.ist.psu.edu/verscheure01joint.html

6. Eager D L，Ferris MC，Vernon MK.: Optimized caching in systems with heterogeneous client populations. J. Performance Evaluation, Special Issue on Internet Performance Modeling, VOl42 (2), 2000，pp:163 -185.
7. Yan R X，Hu Y G，Wang HJ，et al.: Partitioned proxy: a new caching method for video-on-demand system. J. Journal of Northeastern University (Natural Science), Vol23(8),2002,pp:742 -745.
8. Yuqi Hu，Huaiquan Zang, Yuan Gao.: Optimal Proxy Cache Allocation of VOD System. Journal of Northeastern University (natural science), Vol25（4）, 2004, pp: 341-344.
9. Yuxing Peng，Fu Chen.: Video Data Storage of Distributed VOD System Chinese Journal of Computers, Vol23(6),2000, pp:671-672.