A Novel Contents Delivery Network System for Efficiently Resolving Long-tailed User Requests

Junichiro HONMA†, Tomohiro TSUJI†, Yutaka ARAKAWA†, Satoru OKAMOTO†, and Naoaki YAMANAKA†

†Department of Information and Computer Science, Keio University
3–14–1 Hiyoshi, Kohoku, Yokohama, 223–8522 Japan
E-mail: †honma@yamanaka.ics.keio.ac.jp

Abstract  Recently, the delivery frequency of the enormous unpopular High Quality Video contents has been increasing due to the emergence of long-tail phenomena. We propose a new CDN system that uses Surrogate Server (SS) clustering and 2 steps download mechanism. In the proposed system, all contents are classified into three classes based on the number of demands, and different download mechanisms are applied according to their class. The user access time and Original Server processing rate are evaluated by the simulation.

Key words  contents delivery network, long-tail, Original Server processing rate, user access time
1. Introduction

A Contents Delivery Network (CDN) is an intermediate layer that helps efficient delivering of multimedia contents from contents providers to many users [1] [2]. In CDN, Surrogate Servers (SSs) are geographically distributed and each SS has replicas of some popular contents. By resolving user requests at the SS, the load on the Original Server is distributed to SSs. When a user requests a content to the nearest SS and if the SS does not have the requested content, the request is forwarded to another SS. If this SS has the content, the SS serves to the user. And if all SSs do not have the requested content, the user request is forwarded to the Original Server. In this way, CDN system provides the smooth contents delivery to users when users request some popular contents.

In the conventional CDN systems, each SS delivers only the popular contents and the Original Server delivers the rest [3] [4]. Theoretically this CDN system is efficient under the condition that the distribution of the demands is based on Pareto phenomenon [5]. Under this phenomenon, the SS that has the most popular 20% of all contents can resolve 80% of all user requests, and even if the SS that has the most popular 4% of all contents can resolve 64% of all user requests. So the availability of the unpopular contents can be ignorable. However, the Pareto phenomenon that is assumed to be theoretically true is turned out in the current Internet. The contents delivery through the Internet with the growth of E-business changes the phenomenon. Recently, the delivery frequency of the enormous unpopular contents has been increasing. This is called as a long-tail phenomena. The proposed CDN system consists of four mechanisms. They are contents classification, SS clustering, 2 steps download which can be applying different download mechanisms according to the demands of the contents, and policy based cache processing. We explain the proposed system and its effects.

2. Proposed CDN System

The proposed system can be applied under the circumstance that the user requests are controlled by the long-tail phenomena. The proposed CDN system consists of four mechanisms. They are contents classification, SS clustering, 2 steps download which can be applying different download mechanisms according to the demands of the contents, and policy based cache processing. We explain the proposed system and its effects.

2.1 Contents Classification

The popularity of contents is a significant factor for CDN systems. In the proposed CDN system, all contents are classified into three classes (Class1, 2 and 3) based on their popularity. Generally, the contents that belong to the Class1 have a lot of demands. In Fig. 1, the ratio of each Class is A : B : C. In the proposed CDN system, this ratio is set to 0.04 : 0.16 : 0.80 temporarily. In the conventional CDN systems, only class1 and 2 contents are cached at SSs.

2.2 SS Clustering

One of the features of the proposed CDN system is that the SS try to share the contents with other SSs. Since the number of stored contents in the SS cache is limited, SS can’t resolve the all requests. In order to reduce the Original Server processing rate, we introduce the concept of clustering. That means several SSs belong the same group and share contents within the group. When the SS that receives a user request at first can’t resolve it, the SS forwards it to other SSs in the same cluster. To raise the efficiency of the clustering, the SSs in the same cluster have a variety of contents without having same contents. But it isn’t reasonable not to allow more than one SS store the same content, so we allow Class1 contents and Prefix; discussed in 2.3, of Class2 contents are stored more than one SS.

In addition to it, we make the Class3 cluster for the Class3 contents. Just only when the user request of Class3 content occurs and both the SS and the cluster can’t resolve it, the
SS forwards it to the Class3 cluster. The reason why we adopt the additional cluster only for the Class3 contents is that expanding the cluster size causes the increase of user access time but generally the number of Class3 contents is extremely large, therefore to make the enough capacity for Class3 contents we need to expand the cluster size. Figure 2 shows a example of clustering. In Fig. 2, the same pattern of the SS means same cluster and same background color means same Class3 cluster. In Fig. 2, there are 4 SS clusters for Class1 and 2 contents and 3 Class3 clusters. In each cluster, there is only one cluster representative SS that grasps the other SSs’ cache state. SSs forward the search query and know where the objective SS is whenever SS can’t resolve the user request. In addition, there is only one Class3 cluster representative SS in each Class3 cluster. Resolving the user requests in collaboration with several SSs can reduce the Original Server processing rate.

2.3 2 Steps download

We introduce the 2 steps download by using contents classification and SS clustering. At first, we explain the contents splitting. SSs administrate Class2 and 3 contents as a format of Prefix and Suffix as shown in Fig. 3. Prefix is a head block of contents and Suffix is the rest. When downloading a new content, SS stores the full size content in the cache. However, when a new user request comes and it requires to make the storage space for the new content, SS remove the Prefix or Suffix of the stored content in the cache based on policy based cache processing. The detail of policy based cache processing is explained after this section. By storing the contents as Prefix or Suffix, not full size, the number of contents in the cache increases. As a result, the probability that user requests can be resolved in the SS and the cluster increases. Figure 4 shows the flow of 2 steps download. When the user request arrives at the SS, the SS first searches and gets the Prefix of the content, and then does the Suffix during delivering Prefix to the user. After delivering Prefix, it starts to deliver Suffix. This CDN system can provide the delivery of the content by playing Suffix after Prefix seamlessly even if there is no full size content in the cluster.

2.4 Policy Based Cache Processing

Since the capacity of the SS cache is limited, the SS can’t store all the contents in the cache. So, the SS needs to remove some contents in order to make the storage space for a new content after each user requests. In policy based cache processing, three priorities (high, middle, and low) are assigned to each content. As the priority is low, the content tends to be removed from the cache easily. The content with high priority is never removed before all middle and low priority contents removed from the cache. Table 1 shows the priority assignment results of each Class.

If the Class1 content is removed from the cache, the probability that another SS in the same cluster can resolve the user request is thought to be high. Therefore it isn’t needed to set the high priority for the Class1 content. Next, for the Class2 content, we should see to it that the couple of Prefix and Suffix exist in the cluster at least one. We try to have the couple exist in the cluster by considering two cases. Finally, for the Class3 content, only if the only one Prefix of the content exists in the Class3 cluster, the high priority should be set to the Prefix of the Class3 content. When the content has turn to be removed, only the Suffix should be removed. This is because, the number of Class3 contents is
Table 1 Priority assignment of each class.

<table>
<thead>
<tr>
<th>Class</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class1</td>
<td>middle</td>
</tr>
</tbody>
</table>
| Class2  | (a) SS that gets the contents from Original Server  
Prefix : low  
Suffix : high  
(b) Other  
Prefix : middle  
Suffix : low |
| Class3  | (a) When more than one Prefix of the same content doesn’t exist in the Class3 Cluster  
Prefix : high  
Suffix : low  
(b) Other  
Prefix : middle  
Suffix : low |

much more than Class1 and 2 contents, we need to avoid the case that more than one Prefix of the same content exist in Class3 Cluster in order to increase the variety of contents.

2.5 Contents Delivery Mechanism

In the proposed CDN system, the different download mechanisms are applied according to the Class. We describe how to resolve the user request at SSs based on the Class.

2.5.1 Class1

For the Class1 contents, which are popular contents, the Original Server pushes the content to some SSs, and the SSs forward the copy of it to other SSs in the SS cluster. When the user request arrives at a SS, the SS can resolve it without forwarding the user request to the SS cluster in most cases. If the SS doesn’t have the requested content, however, since the probability that other SSs in the cluster have the requested content is quite high, the SS forwards the user request in the SS cluster and gets the content from another SS. So for the Class1 contents, the user access time and Original Server processing rate is sufficiently low.

2.5.2 Class2

When the user request is arrived at a SS, the SS first checks its own contents list to judge whether it has the requested content or not. If not, the SS forwards the user request to a cluster representative SS to know where the requested content is stored in the cluster. When there is no requested content in the cluster, the SS tries to get the content from the Original Server in full size. Meanwhile the case that the requested content exists in the cluster should be divided into 2 cases according to the contents format; full size or the couple of Prefix and Suffix. In the former case the full size content exists in the cluster, the SS forwards it to the user. In the latter case, the SS uses the 2 steps download and gets the content in the order of Prefix to Suffix. In this matter, for the Class2 contents, the Original Server processing rate can be substantially reduced by trying to resolve the content within the cluster as possible.

2.5.3 Class3

As same as the case of Class2, when the user request arrives at the SS, the search is done in order of the SS itself to the SS Cluster. But only the case of Class3 content, the SS additionally tries to search the content in the Class3 cluster. The SS forwards the user request to a Class3 cluster representative SS. If the user request can be resolved within the SS cluster, the rest flow is same as Class2. Otherwise the SS judges whether the Prefix of the content exists in the Class3 cluster or not by the response of the Class3 cluster representative SS. If not, the SS gets the full size content from the Original Server. Meanwhile if there is a Prefix of the content in the Class3 cluster, the SS tries to get the content in the order of Prefix to Suffix that is gotten from the Original Server. For the Class3 content, the Original Server processing rate can’t be reduced as much as the case of Class2 content because the frequency that the user request is resolved only within the cluster is low. However, we expect the improvement of the performance by introducing the Class3 cluster whose range to search is wide.

3. Performance Evaluation

3.1 Simulation Model

We compare our proposed CDN system with the conventional CDN system that each SS has only the popular contents. We consider a mesh network as shown in Fig. 5. The mesh network is consisted of SSs, and an Original Server is independently located out of this mesh topology. The Original Server is directory connected to several SSs. The connected SSs are selected randomly. Total number of contents is 1000. All contents are classified into three classes. The proportions of Class1, 2 and 3 are 40:160:800. The simulation parameters are shown in Table 2. As the generation pattern of user requests, the data of the Oricon music delivery in 2005 is used. We define user access time as a time period from the SS receiving the user request to starting deliver the content. And we also define offered load as the number of user requests generated in the whole network per 1 sec. In this simulation, when the user request can’t be resolved due to the traffic congestion, resending is done after 0.8 msec. Even if the user request can’t be resolved after 30 times resending, it is aborted.

3.2 Performance Evaluation

Figure 6 shows the Original Server processing rate versus the popularity ranking of the content. This was done at the offered load is 2.8 requests/sec. It shows that our proposed CDN system offers lower Original Server processing rate than the conventional CDN system in the case of Class2 and Class3. In the case of Class2, Original Server process-
The shortening of the user access time directly has an effect on the users’ stress, the effectiveness of the proposed CDN system is surely big. Figure 7 also shows that our proposed CDN system needs more access time when it delivers the Class1 contents. In this case, however, this can’t be the factor affecting the users’ stress because the increased access time is small enough as a human feeling.

Figure 8 shows the offered load versus the user access time. In this evaluation, we introduce the concept of the Contents Classification to the conventional CDN system, which doesn’t have that concept basically, in order to simply comparing...
the proposed CDN system with the conventional CDN system. It shows that although the proposed CDN system needs more user access time when it delivers the Class1 contents, it can reduce the user access time substantially compared to the conventional CDN system when it delivers the Class2 and 3 contents. In the conventional CDN system, the user access time of the Class2 and 3 contents increase with the increase of the offered load. This reason is as follows. As the offered load increase, a number of user requests are forwarded to the Original Server and the bandwidth between the Original Server and SSs is consumed. Therefore request resending often occurs and user access time would increase. In Figure 8, we can see that the user access time is stable when the offered load is over about 3 requests/sec in the conventional CDN system. In these offered loads, since the congestions are occurred at the links around the Original Sever, many user requests are aborted because the number of resending exceeds the limit, 30 times. It is thought that the number of aborted user requests would be increase exponentially. While in the proposed CDN system, although the increasing tendency can be seen with the increase of offered loads, the increase of user access time is restrained by resolving user request within the clusters. That is, it is proved that the proposed CDN system is effective in any offered loads.

4. Conclusion

In this paper, We have proposed a novel CDN system that applies different download mechanisms according to the popularity of the contents. The proposed system consists of contents classification, SS (Surrogate Server) clustering, 2 steps download, and policy based cache processing.

The performance of our proposed scheme is evaluated by computer simulations. As a result, we shows that the Original Server processing rate of unpopular contents is reduced by $1/5 \sim 1/30$ and the user access time is reduced by about $1/5$.

References