BS-10-5

Supporting Strong Web Consistency in Content Delivery Networks

Zhou SU　Hongyu GU　Jiro KATTO　Yasuhiko YASUDA

School of Science and Engineering, Waseda University

1. Introduction

CDN (Content Distribution Networks) has emerged as a solution to improve the network load and user latency by replicating the copies of web contents on a group of geographically distributed servers. However, an important issue that still needs to be resolved is consistency management [2], which means the cached copies on geographically distributed servers must be updated if the originals change.

Since using inconsistent (stale) copy to serve user requests is undesirable, the CDN should ensure a strong Web consistency [1][3]: the stale version of the modified document should not be returned to end-users.

In this paper, we therefore talk about how to support a strong Web consistency in CDN. We firstly carry out theoretical analyses of Web access distribution, content server relationship and network traffic. Based on these analyses we then propose a novel algorithm in which not only the access distribution of Web contents but also the current network topology are considered. Through simulations, we check the performance of our proposal when the related parameters are changed. Simulation results show that our proposal can efficiently minimize user delay and network traffic.

2. Previous Work

In *Propagation* method, the updated version of a document is delivered to all copies whenever a change is made to the document at the origin server. Although the copies always keep the latest version of the originals by the *Propagation*, this method may generate significant levels of unnecessary traffic if documents are updated more frequently than accessed.

Another conventional method is *Invalidation*, in which an invalidation message is sent to all copies when a document is changed at the origin server. This method does not make full use of the distribution network for content delivery and each replica needs to fetch an updated version individually at a later time. Therefore, the user-delay may get worse if a frequently accessed document can not be updated at time.

3. Proposal

3.1 Parameter Definition

We assume that each content server is located in a different administrative domain, such as autonomous system (AS). Let λ_i (bytes/second) denote an aggregate request rate from clients to the server i (i={1,…,I}). We assume that there are J documents in our CDN. A parameter $P_{i,j}$ (j={1,…,J}) is defined as request probability for document j from the AS i and $X_{i,j}$ is denoted to be a parameter which takes a binary value of

$X_{i,j}$ =1 (if a copy of the document j is stored at server i)　(1)

　0 (otherwise)

We also define a matrix $\boldsymbol{X}$ of which element is $X_{i,j}$.

3.2 Proposed Algorithm

3.2.1 Access-Based Selection

The first step of the proposed algorithm is introduced in this section. When a document j changes on its original server and $X_{i,j}$ =1 (which means there is a copy of document j on content server i), a threshold $Th1$ will be set up.

If $P_{i,j} > Th1$　(2)

The update process on content server i will be carried out according to the procedure in Section 3.2.2.

Otherwise

The update process on content server i will be carried out according to the procedure in Section 3.2.3.

For example, In Figure 1 there are 5 content servers where copies of document j exist. According to the above selection, content servers will be divided into two groups: content server 1,2,3,4 will go to Section 3.2.2, and content server 5 will go to Section 3.2.3.

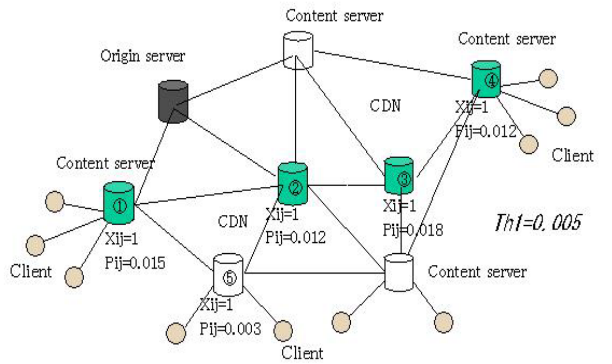

Fig 1. Access-Based Selection

3.2.2 Shortest Path Update

In the conventional method, if a content server decides to fetch the latest version of a modified document, it will contact the modified document's original server of to fetch its latest version. However, as all fetching requests are sent to the same original server, this method causes poor user-delay and load balance. Some researchers also advocate fetching the latest version from other content server instead of the original server, BS-10-5

however, how to select a proper content server is still a problem to be dealt with.

In this subsection, we made the analysis of network traffic cause by sending the modified version of a document. Our goal is to take the latest version of the modified document from nearest content server to minimize the the average AS Hops.

Assume that document j is originally stored in server $o(j)$. Then, if a client sends a request for document to server $o(j)$ via a certain server k, the hop counts during the delivery from server k to server $o(j)$ is given by

$$A_{k,o(j)}=(1/\sum_i\lambda_i)\cdot\lambda_k\cdot P_{k,j}\cdot D_{k,o(j)}(\boldsymbol{X}_0) \quad (3)$$

$D_{k,o(j)}(\boldsymbol{X}_0)$ is the shortest distance from server k to server $o(j)$ under the initial placement pattern $\boldsymbol{X}_0$ and $\Lambda=\sum_i\lambda_i$ is the total request rate from all the domains.

It can also obtain the traversed AS hops if server k fetches the latest version of document j from other content servers q where $X_{q,j}=1$.

$$A_{k,q}=(1/\sum_i\lambda_i)\cdot\lambda_k\cdot P_{k,j}\cdot D_{k,q}(\boldsymbol{X}_0) \quad (4)$$

If a server k is selected to go to section 3.2.2, It will compare $A_{k,o(j)}$ with other $A_{k,q}$ ($X_{q,j}=1$, $q=\{1,\ldots,J\}$) and fetch the latest version of the modified document j from the server with the lowest A.

3.2.3 Adaptive Invalidation

As we have introduced in 3.2.1, If $P_{i,j} < Th1$, the content server will not go to the process in 3.2.2 but go to the process in 3.2.3. In this subsection, when a document is changed at the origin server, only an invalidation message is sent to these content severs. We think the method in this section is a kind of adaptive *Invalidation*, which is different from the conventional *Invalidation*, because *Th1* is set up and it can be changed (adaptive) based on various network situation.

4. Simulation Results

Recent studies show that most communication networks have Power-Law link distributions, where the i'th most connected node has Ω/r_i^{β} neighbors, the network topology in the simulation is decided by the Power-Law link distribution with $\beta=0.8$. The request distribution yields to a Zipf distribution with a Zipf parameter of *0.8*. Client requests arrive according to the Poisson process. There are *1000* documents in CDN and *Th1* is set to be 0.005. All clients are always redirected to the closest server. The total request times in the simulations are *100000*.

There are three replication algorithms we will study:

- *Propagation*
- *Invalidation*
- *Proposal*

Figure 1 shows the result of Old hit, which means that the requested data is not of the new version. Because the updated version of a document is delivered to all copies whenever a change is made to the document at the origin server in the *Propagation*, the old hit of it is zero. However, the network traffic caused by the *Propagation* is very serious in Fig.2, where the traffic caused by sending new version of the modified document is shown. As for the *Invalidation*, although its traffic is the least, its old hit is the worst. Our proposal can get the balance of two conventional ones: Compared with the *Invalidation*, their traffics are closed to each other but its Old hit of our proposal is much better. Compared with the *Propagation*, although the old hit of our proposal is a bit more than the *Propagation*, its traffic can be greatly reduced.

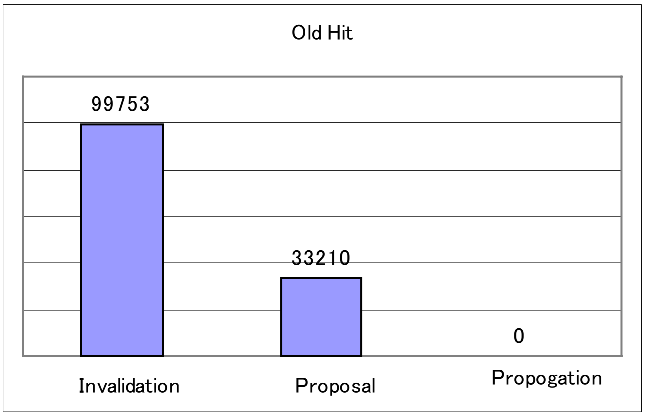

Fig.1: Comparison of Old Hit with Different Replication Algorithms.

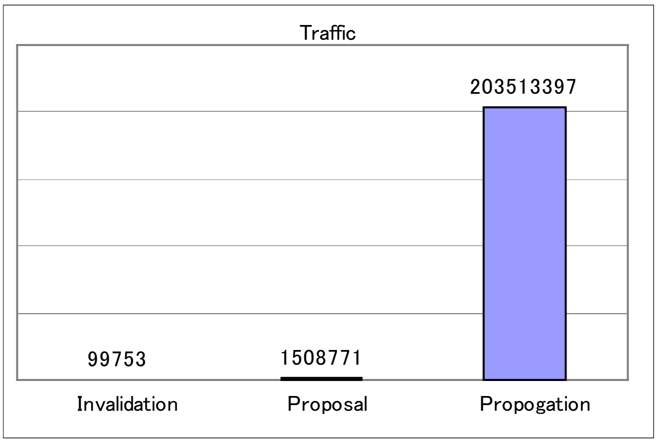

Fig. 2: Comparison of Network Traffic with Different Replication Algorithms.

5. Conclusion

In this paper, we proposed a novel algorithm to support a strong Web consistency in CDN servers based on theoretical analyses of Web access distribution and network traffic. Simulation results showed that the proposed algorithm could obtain better performance than other conventional ones.

References

[1] Zongming Fei,"A Novel Approach to Managing Consistency in Content Distribution Networks", Proceedings of the Sixth International Workshop on WebCaching and Content Distribution, Massachusetts, USA, June 20-22, 2001

[2] Pei Cao and Chengjie Liu 、"Maintaining Strong Cache Consistency in the World Wide Web"、IEEE Trans on Computers, Vol. 47, No.4, April 1998

[3] Venkata Duvvuri, Prashant Shenoy and Renu Tewari、 "Adaptive Leases: A Strong Consistency Mechanism for the World Wide Web "、IEEE Trans on Knowledge and Data Engineering, Vol.15, No.5, Sep/Oct 2003