

# Simple Usage-Based Charging of Web Cache Services

K.G. Anagnostakis, F.C. Harmantzis  
{kanag,harma}@csd.uoh.gr<sup>1</sup>

Computer Science Department, School of Sciences, University of Crete  
P.O. Box 2208, 71409 Heraklion, Crete, GREECE

## Abstract

*Web Cache Services are essential for the survival and success of the Internet. Service providers must possess means of offering this service on a charged basis, as these services are beneficial both to them and their customers. The nature of Web cache traffic implies that to be fair and effective, charging should be based upon higher-level accounting facilities, rather than simple network layer byte counters. We propose a charging scheme that splits the cost of each cached object unequally among the users that requested it, according to measured delay or based on simple hit-miss information. In this way, users are given incentive to always use the cache, bringing new information and requesting already fetched objects from the cache, and cache service providers are able to at least recover costs or even make profit.*

## 1. Introduction

The Web has been the driving force of Internet evolution since its introduction in 1992 [1]. Its multimedia capabilities as well as the semantic power of hyper-links has realized a unique global distributed information system. Network service providers and content service providers are now in a continuous race of upgrading trunks and installing powerful servers to keep up with the users' demand in bandwidth and computational power of information servers. For the Web not to become the victim of its own success engineers soon realized the need for effective replication, distribution and caching of information in order to relieve links and servers from congestion. Caching proxies [2] are intermediate components that extend the traditional client-server model in order to take advantage of both temporal and spatial locality observed in Web accesses with lots of improvements been added to provide efficient caching. With the continuing commercialization of the Internet, the introduction of usage-based pricing and the ongoing research in Internet economics [3], some issues are raised regarding Web cache services. Due to the lack of adequate schemes to charge for cache services, many providers which could offer caching as a complementary service to network connectivity, do not offer or are reluctant to the idea of offering such services, assuming that corporate users will manage to install caches for themselves. Furthermore, they do not possess means of effectively measuring resource consumption through their caches. Knowing that Web traffic dominates in traffic statistics, this issue becomes extremely critical. As discussed in [8], charging can affect patterns of use and hence act as a weapon in demand management. If a service is perceived to be free, there are no constraints on its use. If the same service is offered through two different mechanisms, with more or less different performance characteristics, charging can affect the users' strategy. Charging can encourage users to adopt patterns of use which increase social welfare. On the other hand, there is certainly the need to achieve redistribution of funds saved through caching. Generally, it seems fair that charges should reflect costs, a property which should hold for a fair charging scheme.

## 2. Towards a charging scheme for Web cache services

Service level pricing is required in cases where services have semantics that cannot be left out of a pricing scheme and that cannot be captured by network level accounting mechanisms. In [4] for example, a pricing policy for scaleable VoD applications is described which utilizes service level

---

<sup>1</sup> The authors are also with the Telecommunications & Networks Group, ICS-FORTH

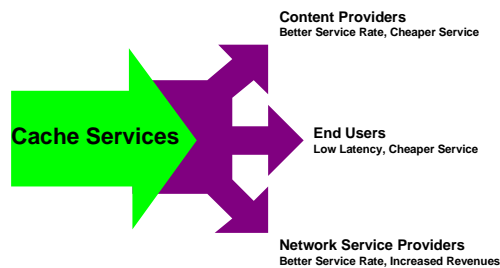
information to charge users in a way that affects user's choices and increases social welfare. Web caches are another case, where hits and misses have different semantics which cannot be left out of a fair pricing scheme, and which cannot be easily identified by TCP and IP level accounting schemes. A theoretical explanation that is closely related to this issue is discussed in [6], where information provisioning infrastructures are characterized based on application and content awareness. Where IP level connectivity is offered, the architecture of the Internet is regarded as application-blind, as applications are not visible to the network service providers. In the case where network service providers offer higher, application/middleware-level services, such as caching, the service architecture becomes application-aware.

From the users' point of view, the cache does not always improve the perceived service quality. For some, the cache offers an increased response time in terms of page retrieval delay, in contrast to the delay of retrieving objects through potentially congested wide-area links. A large portion of the users, access the network through low-speed dial-up connections. For those, the bottleneck is the dial-up connection, in which case, the cache doesn't offer any improvement to service quality.

So, to make caches attractive to users, one can offer discounts for connections to the cache. However, a flat discount policy raises some objections. It does not reflect actual resource consumption, as different documents are more frequently accessed than others. Different users behave in different ways. The cache does not possess methods of controlling its budget and there is a certain risk of building a lossy cache business.

While the above objections could be overseen in a monopolistic environment, in a competitive market, users would surely be interested in being charged in a fair manner. Furthermore, in the case where operators wish to measure intra-organizational resource consumption, rather than charging real money, this scheme is problematic. Therefore, one has to take advantage of accounting information generated by the cache mechanism to reach a reasonable pricing scheme. Publicly available software for running cache servers provide detailed accounting records per request, including time-stamps, requesting host, object location and byte count. This can be used to effectively charge for Web cache services in a per-object fashion. In this document, we present properties, realization and effects of such an algorithm, and discuss directions for further work.

## 2.1 The Information Market



**Figure 1. The impact of caching**

A pricing scheme for Web cache traffic should be able to identify individual users as well as institutions. In this way it should work well for both regional caches that want to recover costs from customers that are institutions and for campus-wide caches where network managers are interested in having a view of contribution of users to resource consumption. The cache should offer better price/performance to the End User to be competitive. For each stakeholder, the charging scheme should have the following effect:

- The Network Service Provider is happy if its networks are less congested and thus deliver better quality to the end users.
- The Cache Service Provider should possess a profitable business, if this service is offered on a commercial basis. If cache services are offered by the Network Service Provider, revenues are affected

Traditionally, there are three stakeholders involved in the Information Market: The End User, the Network Service Provider and the Content Provider. We add another role to the market: the Cache Service Provider. This is similar to the magazine and newspaper distribution system: if someone wants to buy a magazine he does not order it from the company that produces it, he can buy it from his local newsstand instead. The cache plays the role of the newsstand, which manages to have popular magazines in-store for the mutual benefit of all involved parties. One major difference in favor of electronic publishing is that the newsstand is able to reproduce and resell by itself and does not have to order multiple copies of an object.

in two ways: direct profit from the cache and indirect as described above. Additionally, except of cost-sharing, Cache Service Providers should also recover operational and maintenance costs.

- The Content Providers enjoy less loaded servers that can thus satisfy more requests and with better quality.
- The End Users, under conditions, enjoy better service quality in terms of delay, cheaper service, or both, which generally means that the price/performance ratio is certainly improved. The End Users are happy if the cache offers them reduced delay and also saves them money.

In this document we regard the problem only at the user/cache interface, leaving the other issues to be further discussed.

## 2.2 Cache Dynamics

Some interesting remarks can be drawn from analyzing cache access patterns. There has been a lot of work in the area of Web access analysis, especially while seeking strategies and algorithms for effective prefetching. In [7], a user's inter-request time, which corresponds to the user's thinking time is modeled as a Pareto distribution and self-similarity in web access patterns is being discussed. An interesting observation is that different categories of users can be identified based on their behavior. Users exploring the cyberspace, sometimes called "net-surfers" behave differently than users that explore what a document can offer to them, sometimes called "conservative". According to a survey [9], the Web is mainly used for browsing (79.0%) followed by entertainment (63.6%) and work (51.8%). We expect each of these categories to have different request characteristics, mainly in terms of distribution of object popularity.

We have studied cache access patterns at our local Web cache, which serves the ICS-FORTH campus, for a period of a few days. Objects are grouped into classes, according to how many times an object was requested, in other words, the popularity of objects.

This information gives a more detailed insight into access patterns and is the basic input to our pricing model. The classification of objects gives a completely different picture than simple cache hit/miss calculation. We see that objects per class are reduced very fast as the class number increases. Class 1 objects are the well-known cache misses that bring a document to the cache but which are not re-referenced in the future. A further analysis of cache access patterns shows that

Class	Objects	Traffic	Class	Objects	Traffic
1	10994	132733474	13	17	494429
2	1949	30756174	14	17	1028034
3	629	21457050	20	5	1487540
4	304	17843096	21	12	1031751
5	163	5581510	22	15	1035518
6	137	4587660	29	5	1386026
7	85	1921262	42	2	1392300
8	59	2329088	78	2	2894502
9	49	931410	79	2	1157429
10	58	1112870	130	1	259090
11	24	674652	523	2	1368168
12	19	548184			

Table 1.

users exhibit different class distributions of objects requested, which agrees with the above mentioned diversity in the user population's expectations from Web services.

## 2.3. Dealing with cache-meshes

The problem of sharing the cost of an object in a mesh of caches is similar to the problem of cost-sharing in multicast trees. The difference lies in the way and time information is distributed to users. In multicast transmission, all users participate in the session simultaneously, which has the advantage of knowing the exact number of users that have to share the cost of the multicast tree. In a cache mesh, a tree exist for each object. We regard the root of that tree to be the cache server that "hits", the tree being populated by caches that were involved in accesses for the object and the users being the leaves. An analysis of cost-sharing of multicast trees is provided in [9], which discusses several aspects of the problem in depth. In this work, we will only deal with a single cache. Issues of charging in cache-meshes will be discussed in a forthcoming paper.

## 2.4. Desired Properties for a Charging Algorithm

Considering the results of the analysis of cache access patterns, and the stakeholders' interests stated above, we can specify properties that a charging scheme should meet:

- The cache should be able to charge the user, as soon as possible; ideally, after a request is serviced. This, while being reasonable, is not an easy task: the cache does not know and cannot easily predict the cost of fetching the object from the remote server and also does not know the class it belongs to. Alternatively, the charge can be calculated at the end of the object's life-cycle, that is, the time at which the object is removed or updated. The last option is to perform on-line calculation of an upper bound on the charge that could be sent to the user at the time of the request. A pricing scheme should ensure that upper bound in all cases.
- For an object in class 1, the price should be less than or equal to the price the cache had to pay to fetch it from the remote server. This is to motivate the users, by trying to set the price as low as possible. If the price is equal, there is no incentive but also no reason to use or not to use the cache. Naturally, if the price is higher, users would not be interested in using the cache. According to this, the cache may have to take a risk by setting a lower price, hoping to recover the cost from future accesses.
- If the algorithm calculates charges at the end of an object's life-cycle, cost recovery can be achieved from other objects, possibly from objects belonging to classes other than the first, the algorithm having to keep state on overall and relative benefit/loss. The price of objects belonging to these other classes, should be set in the same manner, respecting the offered service quality.
- Users accessing objects that belong to class  $N$ ,  $N > 1$  may enjoy a better service quality in terms of object retrieval delay, as they receive the document from the cache. This could imply that the tariff for them should be greater than that for the first user accessing that object, who suffers the relatively lower service quality.
- Users that are serviced through cache hits, could be set a higher tariff than the price of fetching the document from the server. While this seems fair, one cannot know if the user cares about service quality and whether he it is in his interest or in his capabilities to pay more to get an object faster. Thus, our policy must be clear in this case: the cost of fetching an object from the cache is lower or equal to the cost of fetching the object from the remote server, regardless of object class and time of access.

## 3. Proposed Charging Algorithm

Let  $c_S^j$  be the cost for cache  $S$  to fetch object  $j$  from the remote server. The charge for user  $i$  is  $c_i^j$  and  $1 \leq i \leq N_j$ , where  $N_j$  is the number of users that requested object  $j$ . The sum of charges for object  $j$  is  $a_j = \sum_{i=1}^{i=N_j} c_i^j$ . Generally,  $c_i^j = \frac{c_S^j}{N_j} w$ , which reflects the general idea that the cost should be shared by the  $N_j$  users accessing the object. In the following paragraphs we will describe and compare strategies for selecting parameter  $w$ , their properties and impact on the cache economy.

### 3.1 Simple Cost-Recovery Scheme with Miss Incentive

We want to provide a reward to the user's that cause cache misses and bring new objects to the cache. The first user to fetch the document, causing a cache miss, will be charged  $c_A^j$ . The remaining  $N_j - 1$  users that enjoy a cache hit will be charged  $c_A^j$ . The sum of the charges is:

$$a_j = c_A^j + (N_j - 1)c_A^j \quad (3.1)$$

The first user is charged:

$$c_A^j = \frac{c_S^j}{N_j} w_j \quad (3.2)$$

where  $0 < w_j < 1$  determines the level of discount to the user that caused the miss. The discount to users that suffered a miss is justified for non-class-1 objects, due to the relatively higher delay thus lower service quality, which the remaining requesting users do not experience. The remaining users are charged:

$$c_A^j = \frac{c_S^j}{N_j} w_j' \quad (3.3)$$

Let us see what happens if we want to have zero-profit by the cache. From (3.1), we see that for this to happen,  $a_j = c_S^j$  must hold. So we have:

$$c_S^j = \frac{c_S^j}{N_j} (w_j + (N_j - 1)w_j') \quad \text{where } w_j + (N_j - 1)w_j' = N_j \quad (3.4)$$

This means that, while for objects of class 2 and higher, the cache is able to recover costs, by solving (3.4) for  $w_j'$ , if  $N_j = 1$  then  $w_j = 1$  and there is no discount for class 1 objects. This seems natural, and through simulations, has been proven to be necessary for effective budget control. For example, we have identified people that use the cache for retrieving information that is destined to them only, which means that there is by definition no chance of sharing this information with other people. In this case, there is no meaning in providing a discount to them. While this is the extreme case, in the average case where a user has a certain object/class distribution, the user will experience a discount from non-class-1 objects, which means that the above scheme is incentive compatible.

### 3.2 Cost-Recovery and fixed reward with Miss Incentive

Generalizing, our scheme would benefit from users in all classes except the first one. To formulate this property, we define:

$$a_j = g_j c_S^j \quad \text{where } w_j + (N_j - 1)w_j' = g_j N_j \quad (3.5)$$

The cache should select appropriate values for  $w_j$  and  $w_j'$ , depending on it's policy. As in the previous scheme, if  $N_j = 1$  and  $w_j < 1$  then  $g_j < 1$  and the cache experiences loss. The cache can make up the loss and further make profit by selecting appropriate values for  $w_j'$ . According to this, to make a benefit of  $r_j$ , the cache should set:

$$w_j' = \frac{\left( \frac{c_S^j + r_j}{c_S^j} \right) N_j - w_j}{N_j - 1} \quad (3.6)$$

We see that by selecting appropriate values for  $r_j$  the cache can have full control over it's budget. For example, if a cache advertises discount prices of 20 % (that is,  $w_j = 0.8$ ), and wants to recover twice the cost ( $r_j = c_S^j$ ), an object with a cost of 100 cents which was accessed 4 times will have the following charge on the users: the first user pays 20 cents and the remaining three 60 cents. The cache will get 200 cents from the users and pay 100 to the network provider, making the benefit 100 cents. If the same object were only accessed once, the user would be charged 80 cents and the cache would have to recover from other objects.

This scheme overcomes the previous scheme's problem of experiencing loss because of discount to class 1 objects. The cache is able to cover the loss and further benefit, by adding a small and also fair amount to objects of other classes.

### 3.3 Adding proportional reward

Up to now, we have examined how quality parameters can be incorporated into the charging scheme and how the cache can recover and further make profit from the cache business. The quality parameters can be calculated based on accounting information that is available. However, we will simplify the model by viewing two service categories (hits and misses) and relaxing the requirement for quality-proportional charging. Currently, it is more important to us to ensure that the cache can recover operational costs including the network cost of servicing cache hits and efficiently control its reward. We identify two cases: if an object is accessed only once, then  $w_j$  is defined as above. If an object is accessed more than once, then we define (with  $r_j=0$ ):

$$W_j = w_j + (N_j - 1)p_j \text{ and } W'_j = w'_j + (N_j - 1)p_j \quad (3.7),(3.8)$$

Now, (3.1) and (3.2) are changed to:

$$c_A^j = \frac{c_S^j}{N_j} W_j \quad \text{and} \quad c_A^j = \frac{c_S^j}{N_j} W'_j \quad (3.9), (3.10)$$

Note that for large  $N$ 's, the charge approaches the cost of servicing the request. This observation deserves a second thought: very popular objects, or even statically cached or mirrored objects, can be the source of benefit for the cache provider, as the portion of the charge that reflects the sharing of the cost of fetching the object from the remote server is near zero, and the user pay only the amount the cache wants for servicing the request.

### 3.4 Charging for service quality

A third, slightly complicated variation of the charging scheme involves pure delay-based charging. The current version of "squid" keeps track of the time between the request and the delivery of the requested object to the client. This means that accounting information is available to allow for delay-based charging. Each user pays a fraction of the cost that depends on the quality (in terms of delay) that he enjoys. In this case, we define a quality indicator  $q_i^j$  for each object and requesting user, and:

$$c_i^j = \frac{q_i^j}{\sum_{i=1}^{N_j} q_i^j} c_S^j \quad (3.11)$$

We made two simplifying assumptions here: quality-of-service is regarded as linear to delay and the user's willingness to pay is also regarded as being linear to the received quality-of-service. These assumptions certainly don't apply always. A hybrid scheme, which considers delay but in a more constrained way in terms of lower and upper bound on charge, could be of help in this case. If we want to charge on a quality-of-service basis, we define  $0 < q_j < 1$  to be the relative QoS. We define

$w_j = b_t + (1 - b_t)q_j$  where  $b_t$  is the risk factor, which depends on the cache's budget and policy. It is clear again that if  $N_j = 1, b_t < 1, q_j < 1$  then in (3.5)  $g_j < 1$  and the cache experiences loss but mechanisms to at least recover the loss have been described above. A more in-depth analysis of delay-based charging will be provided in a forthcoming paper.

## 4. Implementation and Experiments

There are some issues regarding the implementation and practical use of our proposed charging scheme. State of the art Web Cache software comes with integrated accounting functionality. Accounting records include date/time, requested URL and file size. Given a network charging scheme, one can calculate the cost as a function of the above accounting information. Otherwise, we have to add charging information to the records held by the cache. There is also the need for mapping the recorded requesting host to a chargeable entity (user, department, institution etc). Static mapping can be implemented through configuration file while dynamic mapping can be implemented by means of a

registration/de-registration procedure similar to the login/logout procedure for using a workstation. Java applets can be employed to monitor the current budget for both users and the cache manager and to implement the registration/de-registration procedure. The charging algorithm can be implemented either as part of the cache server or as an independent tool that parses accounting files. The system can be integrated with billing architectures, such as the one described in [4], to combine cache bills with network service bills.

We have currently implemented the simple charging scheme with proportional benefit and miss incentive as an independent tool that parses cache log files and calculates charges on a per-host basis. We have used the implemented algorithm to calculate the charges of the cache accesses shown in section 2. The results, for a variety of parameters, are shown in table 3, while some basic characteristics of the workload (hit/miss objects and hit/miss bytes) are shown in table 2.

<b>Hits</b>	<b>14318</b>	<b>Misses</b>	<b>14661</b>
<b>Bytes</b>	<b>79590256</b>	<b>Bytes</b>	<b>163243784</b>

**Table 2. Workload Statistics**

Hit $w_j$	Miss $w_j$	$p_j$	Miss Loss	Hit reward	Benefit
<b>0.7</b>	<b>0.75</b>	<b>0.1</b>	<b>66077352</b>	<b>16349161</b>	<b>-49728544</b>
<b>0.7</b>	<b>0.8</b>	<b>0.2</b>	<b>52864302</b>	<b>32717405</b>	<b>-20146897</b>
<b>0.7</b>	<b>0.85</b>	<b>0.2</b>	<b>39652236</b>	“	<b>-6934831</b>
<b>0.7</b>	<b>0.85</b>	<b>0.3</b>	“	<b>49085529</b>	<b>9432820</b>
<b>0.7</b>	<b>0.9</b>	<b>0.1</b>	<b>26437854</b>	<b>16349161</b>	<b>-10088693</b>
<b>0.7</b>	<b>0.9</b>	<b>0.2</b>	“	<b>32717405</b>	<b>6279551</b>
<b>0.7</b>	<b>0.9</b>	<b>0.3</b>	“	<b>49085529</b>	<b>22647675</b>
<b>0.7</b>	<b>0.95</b>	<b>0.1</b>	<b>13224768</b>	<b>16349161</b>	<b>3124040</b>
<b>0.7</b>	<b>0.95</b>	<b>0.2</b>	“	<b>32717405</b>	<b>19492637</b>
<b>0.7</b>	<b>0.95</b>	<b>0.3</b>	“	<b>49085529</b>	<b>35860761</b>
<b>0.7</b>	<b>1.0</b>	<b>0.1</b>	<b>0</b>	<b>16349161</b>	<b>16349161</b>
<b>0.7</b>	<b>1.0</b>	<b>0.2</b>	“	<b>32717405</b>	<b>32717405</b>

**Table 3. Experiment Results**

## 5. Summary

Charging caches will motivate users to use them and providers/network managers to offer them either as complementary service to network connectivity or as an independent service that can be proved highly profitable. From the economic perspective, we see that we can save network cost by installing caches and we can distribute these savings among all stakeholders of the Information Market. We certainly did not rediscover the benefits of using caches: we just map them directly to economic quantities and provide clear policies that users as well as people in the networking business can naturally perceive. A charging scheme, such as the one described in this paper, gives a clear message to corporate users to use caches, to network managers to install them and to the software engineering and research community to improve them.

As we have seen, Content Providers also benefit from cache services but do not actively appear in the proposed scheme. Schemes could be developed that force Content Providers to pay (contract) caches to hold their information (to automate replication and Web-hosting services). The problem of accounting for site accesses through special (third party) access counters is still under consideration. Interesting questions such as inter-cache charging in a market of (hierarchically structured) cache providers and competition between caches arise and need further discussion. Also, valuable input could be gained by formalizing the problem using queuing theory and game theory and by further evaluating work towards Web traffic modeling.

## References

- [1] T.Werners-Lee, R.Caillian,J-F.Groff and B.Pollerman: "World-Wide Web: The Information Universe", Electronic Networking: Research, Applications and Policy, 1(2):52-58, Spring 1992
- [2] A.Luotonen and K.Altis: "World-Wide Web Proxies", Computer Networks and ISDN systems, First International Conference on the World-Wide Web, Elsevier Science BV, 1994
- [3] J.K.MacKie-Mason, H.R.Varian: "Economic FAQs about the Internet", Journal of Economic Perspectives, 8(3)
- [4] R.J.Edel, N.McKeown, P.P.Varaiya: "Billing Users for TCP", Tech.Rep., Dept. of EE and CS, University of California, Berkeley
- [5] A.Krishnamurthy, T.D.C. Little, D.Castanon: "A Pricing Policy for Scalable VOD Applications"
- [6] J.MacKie-Mason, S.Shenker: "Network Architecture and Content Provision: An Economic Analysis", Jun. 1996
- [7] C.R.Cunha, C.F.B.Jaccoud: "Determining WWW User's Next Access and Its Application to Pre-fetching", ICC'97, Alexandria, Egypt, 1-3 July, 1997
- [8] M.Tedd: "Some Thoughts on Charging", Pricing Internet Resources Meeting, April 1996
- [9] S.Herzog, S.Shenker: "Sharing the Cost of Multicast Trees: An Axiomatic Analysis" ,ACM SIGCOMM'95, Aug. '95, Cambridge, MA, USA