Improving the Effectiveness of Web Caching at the Client's Site

Safia Abul-Rub* and Sami Serhan**

Received on April 14, 2003
Accepted for publication on Aug. 28, 2005

Abstract

In this paper, the performance is analyzed and evaluated in terms of hit ratio, wasted bandwidth, network traffic, and user latency under various situations, levels and architectures.

We propose configuration of a WWW caching system that would give an optimal performance of 84% hit ratio, 4.7E+9 bytes of wasted bandwidth, 6.8E+9 bytes of network traffic and 7 seconds of user latency. This configuration suggests the suitability of fully associative mapping cache memory organization at both the client's and the proxy's sites, least recently used replacement policy at the client's site and least frequently used policy at the proxy's cache.

Key words: Web caching, World Wide Web, cache memory organization, cache fetch policy, cache replacement policy, proxy servers, clients, hit ration, wasted bandwidth, network traffic, user latency.

Introduction

Recent years have witnessed an extensive use of the World Wide Web WWW. On the Internet, traffic jams and bottlenecks have become daily occurrences. The increasing popularity of the WWW led to huge amounts of traffic traveling through the Internet and therefore, an increase in latency. Some popular Web documents may become unobtainable at peak times.

For this reason, latency reduction has been among the primary concerns of the Internet research and development community and has opened a huge area of research [1-2], [4-10].

Caching was a vital solution that could not be overlooked. The term 'Web caching' refers to the incorporation of caches in several sites on the Internet [21]. These sites are typically at the clients and at the proxy servers. The major problem being addressed is managing the various caching policies, strategies and organizations involved in a Web caching system in a way that leads to better performance [15,18]. Various environments

© 2006 by Yarmouk University, Irbid, Jordan.
* Al-Itra Private University, Amman, Jordan.
** University of Jordan, Amman, Jordan.
are considered under which Web caching is implemented. The effects of different structures on the overall performance of the system are studied.

Background And Previous Work

Considerable work has been done on caching Web objects. In [3], [23], a LRU extension is proposed, in which Web objects are greedily removed from the cache in the order of recency of last access until enough space is created for the incoming object. Their proposal aimed at providing a model to deal with the heterogeneity of sizes for Web objects.

The work in [14] has investigated Web pre-fetching from the proxy to the browsers by taking advantage of the idle time between user Web requests and using prediction algorithms to predict the next request submitted by users. They have shown that proxy-side pre-fetching is a promising technique that can reduce user observed latency by over 20%.

In the author’s work among the Swiss Federal Institute of Technology (SPIT) [19], the advantages and drawbacks of active caching (pre-fetching) of Web content are discussed. He made his recommendations that active caching functionality must be employed in every WWW caching proxy. He has shown that up to 29.4% of the cached objects will be invalid if the cache proxy does not use active caching. Among them, 27.3% of the objects will not exist any more and 2.2% will have an updated copy on the corresponding WWW servers. His results showed that proxy caching without active caching saves only 6.3% more from original users’ traffic than proxy caching with active caching functionality.

The work in [11] showed that (30%-50%) of cache hits constitute freshness misses. He proposed Web caching refreshment policies that extend freshness lifetime by selectively validating cache objects upon their expiration. By increasing cache freshness, requested objects are more likely to be fresh and thereby are serviced faster. They demonstrated that 25% of freshness misses can be eliminated by applying a good refreshment policy.

The study in [12] has shown that there is an opportunity to improve user response time from cache by improving the cache consistency mechanism and that when a validation is returned from cache directly, it is returned approximately an order of magnitude faster than an object that requires a consistency check.

The work in [20] proposed an intelligent dynamic caching technique to model document life histories. Their simulation results showed that the frequency of requests for a document, rather than object size, is more relevant to the management of Web caches.

The authors in [24] studied partitioning dynamic pages into classes based on URL patterns and proposed a scheme allowing an application to specify page identification, cacheability, and data dependence. Their experimental results show 20% improvement in cache hit-ratios.
On the other hand, the authors in [13] examined the different approaches to Web cache consistency. They have shown that no certain mechanism prevails but that it is dependant on the highest priority performance aspect in the system, and that a weak cache consistency protocol reduces network bandwidth consumption and server load more than either time-to-live or an invalidation protocol and can be tuned to return stale data less than 5% of the time.

**The Proposed Model**

The assumptions and strategies used in our model can be summarized as follows:

- Maximum proxy cache size is 10 GB
- Maximum client cache size is 2.3 GB
- A time-out occurs at a user-latency greater than 45 seconds. Time-out is defined as the stage when no more delay is accepted and the request has to be re-issued.
- For the cache memory organization and mapping strategy, we considered Fully Associative Mapping (FAM), Direct Mapping (DM), and Set Associative Mapping (SAM).
- The following cache fetching policies were considered:
  - On demand (OD)
  - Pre-fetching based on the links in the Web object (if available) (PL). It is assumed that only 30% of the links are pre-fetched and are selected randomly.
  - Pre-fetching based on the contents of the Web object (PO). It is assumed that 40% of matching Web objects are pre-fetched and are selected randomly.
- In both pre-fetching policies, pre-fetching on miss was applied.
- Several policies have been proposed in the literature for cache replacement policies, but not all have been implemented in this study. In all replacement policies, it is assumed that all removals from the cache are done only upon need and not based on any other time-dependent criterion [23].
- The policies studied are First In First Out (FIFO), Least Recently Used (LRU), Least Frequently Used (LFU), and Random Replacement.
- Wasted bandwidth is defined as the total size of Web objects, which have been fetched at the proxy’s cache but not used by more than 10% of the clients, or the total size of Web objects, which have been fetched at the client’s cache but not used more than once.
- The Time-to-live (TTL) coherence protocol was applied. In TTL protocol, expired objects remain cached at the proxy cache. When a request is made to an expired Web object, an ‘if-modified-since’ inquiry is sent to the Web server, so that a new copy is retrieved only if the object has actually been modified (the copy currently at
the proxy cache is stale). The TTL coherence protocol has been chosen, for its implementation simplicity.

Workloads

The workloads used in this study are classified as follows: Artificial workloads and Real-system workloads

Artificial workloads

The reason behind using artificial workloads is to generate representative access patterns that are possible and significant but are not easily found in real-systems’ traces.

In this work, the following patterns were considered:

• Varying the degree of temporal locality

Temporal locality can be generalized by assuming that when a certain Web object, WO, is referenced at time T and followed by a request sequence RS, it is likely when WO is referenced at time T+t, that RS will also follow.

Artificial workloads were used to vary the degree of the causes of temporal locality and study the effects of these variations on the performance of Web caching.

• Varying the degree of spatial locality

The general description of spatial locality can be stated as follows:

If an access is made to a Web object, WO, and that object was found in location L, then the next R references are likely to be found in the same location L. L could be a client cache, a proxy cache, a peer cache or a Web server.

Real-system workloads

Real-system workloads are essential in order to reflect the reality involved in a Web caching system. The workloads used in this study were collected at different organizations. Access traces were collected from four proxy servers at four different sites (named JU, PU, AS, and ASG) distributed throughout the Internet.

Table 1 summarizes the traces. All traces were taken from one proxy server per system.

Three sites used MS Proxy Server version 2.0 while the fourth used Netscape Proxy Server version 3.5. In all the traces, the initial misses have been included.

The choice of sites was based on providing two separate environments and studying the effects of such environments on the performance of the system. Environments cover requests in a particular field and requests that represent public Internet access in which varying interests (access to different unrelated Web objects) resemble the users of the system.
Improving the Effectiveness of Web Caching at the Client’s Site

The Neutral Environment

We have conducted a total of 16 experiments. All experiments have been built and designed based on one ‘neutral environment’. This environment defines the set of Web caching parameter values that are considered neutral so that when studying a certain factor in any experiment, the study assumes that other factors have their values assigned by that neutral environment. Table 2 defines the neutral environment.

Table 1: real-system workloads

<table>
<thead>
<tr>
<th>Site</th>
<th>Collection Period</th>
<th>No. of Clients</th>
<th>Requests (Millions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JU</td>
<td>Sep 1999 – Apr 2001</td>
<td>150</td>
<td>11.5</td>
</tr>
<tr>
<td>PU</td>
<td>Apr 2000 – Jun 2001</td>
<td>35</td>
<td>9.4</td>
</tr>
<tr>
<td>AS</td>
<td>Dec 2000 – Jun 2001</td>
<td>60</td>
<td>7</td>
</tr>
<tr>
<td>ASG</td>
<td>Nov 1999 – Jul 2001</td>
<td>46</td>
<td>21.4</td>
</tr>
</tbody>
</table>

Table 2: definition of the ‘neutral environment’

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Proxy</th>
<th>Client</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Organization</td>
<td>FAM</td>
<td>FAM</td>
</tr>
<tr>
<td>Fetch Policy</td>
<td>OD</td>
<td>OD</td>
</tr>
<tr>
<td>Replacement Policy</td>
<td>LFU</td>
<td>LRU</td>
</tr>
<tr>
<td>Cache Size</td>
<td>6 GB</td>
<td>800 MB</td>
</tr>
</tbody>
</table>

Results and Analysis

We now present the results obtained from our experiments. Some experiments operated on artificial, others on real-system workloads. We classify the results as follows:

- The influence of Temporal and Spatial Localities

As shown in Figure 1, the hit ratio increases with increasing temporal locality. This is a logical consequence, as more clients require the same Web objects, which are present in the cache.

Spatial locality does not acquire the same gradual increasing behavior as temporal locality. At some stages, decreases in hit ratios followed by increments as spatial locality increases are noted. This irregular behavior is directly related to the nature of location L. Whenever L is a cache, the hit ratio increases, otherwise, it decreases.

For the same reasons addressed in hit ratio, as temporal locality increases, wasted bandwidth, network traffic and user latency all decrease as shown in Figures 2 to 4.

307
- The influence of Proxy Cache Size

Figures 5 to 8 show the results on proxy cache size. The four real-systems at hand are examined the following results and analysis are concluded:

- All four systems tend to stabilize at cache size equal to 9 GB, with minor jumps in hit ratios.

- The consequence of increasing the cache size is more noticeable when traces were taken from sites where the degree of 'similarity in interests' is high.

- When the system's environment itself forces a high degree of variations in interests, even when a document is in cache, it will only be referenced by very few clients if not only by the requesting client. In such environments, the proxy cache size has a slow effect.

Figure 6 examines the effects of proxy cache size on wasted bandwidth.

---

**Figure 1:** The influence of temporal and spatial localities on hit ratio.

**Figure 2:** The influence of temporal and spatial localities on wasted bandwidth.

**Figure 3:** The influence of temporal and spatial localities on network traffic.

**Figure 4:** The influence of temporal and spatial localities on user latency.
Improving the Effectiveness of Web Caching at the Client's Site

In Figure 7, traffic is studied over proxy cache size. It is concluded that the 'similarity of interests' issue is a major environmental factor influencing the amount of traffic.

The influence of proxy cache size on user latency is illustrated in Figure 8.

- The influence of optimizing the Proxy Cache Size

The purpose of this experiment was to study how dominating the stabilizing proxy cache size is. By examining Figures 5 to 8 and analyzing the results, this experiment has taken 8 GB as a stabilizing cache size starting point. The size is forced on 50% of the system's proxy caches. Figures 9 to 12 demonstrate the results of the experiments. The word 'average' is added to describe the y-axes as in this case 50% proxy caches of 8 GB have been taken as a basis and varied the rest of the proxy caches in several cases and taken the average of all. The following is concluded:

- A very high increase in hit ratios reaching almost 50% is noted. This implies the strong effectiveness of proxy cache sizes on a Web caching system having an incremental relationship with hit ratio as shown in Figure 9.

- Only a slight reduction (compared to what was obtained on hit ratio) was observed on wasted bandwidth. Going back to the definition of wasted bandwidth - total size of fetched Web objects not used by more than 10% of clients, it is analyzed that the results of Figure 10 are not surprising. Having more Web documents at the proxy cache (with a larger cache size) does not necessitate having those documents requested by more than 10% of clients and, therefore, in this case, the 'similarity of interests' factor dominates as well. The same applies to network traffic.

- Figure 11 shows considerable reduction in traffic over the network happening. With larger proxy servers' caches, the amount of traffic traveling between proxies and Web servers is reduced.

- The influence of different Memory Organizations at the Proxy and the Client sites

The order in which the rest of the experiments, addressed next, were conducted was different. Under a certain test parameter (memory organization, fetch policy, and replacement policy), the experiments study all different possible value assignments for that parameter on all evaluation criteria (hit ratio, wasted bandwidth, network traffic and user latency). The purpose of following this mechanism is our belief that it would help analyze influences on the tested evaluation criterion in more depth.

The word 'average' was added to all y-axes of the results of these experiments to reflect the fact that, in each case, the average of the four real-systems have been taken. Moreover, in all these systems, whenever a client is marked by value X, it is meant 70% of the clients, the other clients varied and the average of all cases was taken. In addition, whenever a proxy is marked by value Y, it means 100% of proxies, i.e. homogeneous proxies. In this analysis, each criterion is considered separately.

Figure 13 shows the effects of varying the clients' cache memory organization over different proxies' cache memory organizations on hit ratio. Regardless of the proxies'
memory organization, it is noted that FAM performs better than either DM or SAM. However, the best hit ratio was obtained when proxies also have a FAM organization, giving a hit ratio as high as 97%. SAM then followed and last was DM giving the lowest hit ratio in all cases. This is again due to the flexibility of FAM over the other memory organizations. It is also concluded that proxies with DM caches will considerably degrade the performance, offering a maximum average hit ratio of only 48%.

![Figure 5: The influence of proxy cache size on hit ratio.](image1)

![Figure 6: The influence of proxy cache size on wasted bandwidth.](image2)

![Figure 7: The influence of proxy cache size on network traffic.](image3)

![Figure 8: The influence of proxy cache size on user latency.](image4)

The results on wasted bandwidth are shown in Figure 14. And those on network traffic in Figure 15. By observing these results, it is concluded that flexibility is the dominating factor concerning memory organization. The same applies for user latency as shown in Figure 16. The results obtained strongly reject using DM at the proxy’s site.

- The influence of different Fetch Policies at the Proxy and Client sites

Figure 17 shows the influence on hit ratio. It is noted that an OD proxy cache has the worst
performance. This can be explained by stating that proxies serve requests to a considerable number of clients, and those clients do have a minimum degree of temporal locality between them. The use of OD results in many Web objects being removed from cache while still needed and as a side effect, a reduced hit ratio. On the other hand, pre-fetching at the proxy site with both its types perform better. A PO proxy however, gives the best results.

Figure 9: The influence of optimizing proxy cache size on average hit ratio.

Figure 10: The influence of optimizing proxy cache size on average wasted bandwidth.

Figure 11: The influence of optimizing proxy cache size on average network traffic.

Figure 12: The influence of optimizing proxy cache size on average user latency.

Pre-fetching 40% of matching Web objects helps in obtaining high degrees of hit ratio, as these objects will be available to all clients, and it is likely for the clients to access different pre-fetched objects. Again, it appears that a matching contents criterion is better compared to available attached links.

At the client's site, it is observed that clients are at their best performance, when the documents are fetched only upon need, i.e. when using OD. It is concluded that when PL and PO are used, hit ratio is reduced since pre-fetching removes needed documents from the clients' caches. In addition, very low percentage of the links and matching pre-fetched objects is used, as they are not needed. OD is sufficient with a pre-fetching...
proxy, as different clients are able to find their requested objects in a nearby proxy upon their need.

**Figure 13:** The influence of cache memory organization on average hit ratio in a Web caching system.

**Figure 14:** The influence of cache memory organization on average wasted bandwidth in a Web caching system.

**Figure 15:** The influence of cache memory organization on average network traffic in a Web caching system.

**Figure 16:** The influence of cache memory organization on average user latency in a Web caching system.

Figure 18 shows the influence of fetching policies on wasted bandwidth. It is noted that an OD proxy continues to give the worst performance by wasting high percentage of bandwidth. This is because, when a proxy demands a Web object, that demand is a subsequent to a client's request. Other clients are likely to request matching or linked objects but not exactly the same copy present in the cache, which leads to a high degree of wasted bandwidth. Therefore, pre-fetching outweighs OD in its benefits. For the same reasons mentioned when discussing hit ratio, PL then follows providing lower wasted bandwidth and PO gives the lowest. At the client's side, OD is best in reducing wasted bandwidth, followed by PL and then PO.
Improving the Effectiveness of Web Caching at the Client's Site

Figure 17: The influence of cache fetch policies on average hit ratio in a Web caching system.

Figure 18: The influence of cache fetch policies on average wasted bandwidth in a Web caching system.

Figure 19: The influence of cache fetch policies on average network traffic in a Web caching system.

Figure 20: The influence of cache fetch policies on average user latency in a Web caching system.

The influence of different fetch policies on network traffic is shown in Figure 19. With OD clients, traffic was high in both an OD and a PL proxy. The purpose of pre-fetching is to reduce traffic on the long run, so traffic is reduced whenever pre-fetching is applied. However, it is noted that OD clients gave the lowest network traffic. This is because a PO proxy was used and a high number of clients found their requests in the proxies’ caches and kept a copy in theirs, so at some stage, traffic was limited to only that required by the coherence protocol. In general a PO proxy gave the lowest network traffic. It is noted however, that heterogeneity in the clients and proxies pre-fetching policies leads to high network traffic as the amount of bytes traveling is maximized due to first pre-fetching links at the proxy’s cache and then matching Web objects at the client’s cache or vice versa.
Abul-Rab and Serhan

- The influence of different Replacement Policies at the Proxy and Client sites

In this experiment, the effects of different replacement policies are examined. By observing the hit ratio shown in Figure 21, it is concluded that the best performance is obtained when using LFU on the proxy side and LRU at the client's side. Random replacements had extremely low hit ratios. FIFO on the other hand had a somewhat similar effect as that of LRU but at a lower scale. At the proxy's side, frequency has proved to be a better replacement policy than recency.

Recency, on the other hand, was a better policy at the client's site. Since proxies serve requests to multiple clients, the least recently used Web documents are not actually the least wanted documents, but least frequently used documents are. At the client's site, the least recently used documents are those less likely to be referenced in the near future. The results strongly reject random replacement at the proxy side. Moreover, FIFO replacement has not proved to be a performing policy as with increasing locality and some special circumstances, the first Web objects entering the cache are highly requested.

The results obtained on wasted bandwidth and network traffic are shown in Figure 22 and Figure 23 respectively.

The more complex a replacement policy is, the more latency is expected at the clients' side. However, the results obtained on user latency and those shown in Figure 24 suggest that a good replacement policy at the proxy's and clients' sides prove to offer lower user latency on the long run.

- The influence of Homogenous and Heterogeneous Proxies

In all previous experiments, only homogenous proxies were studied at a time. Alternatively, heterogeneous proxies can be used. It is important to examine the effects of incorporating heterogeneous proxies. Heterogeneity can be at various levels including memory organization, fetch policy, replacement policy and prediction strategy. In this experiment, heterogeneity was varied at one level at a time and the average of all has been taken. A strong recommendation of homogeneous proxies is concluded.

The further complexity that heterogeneity creates, results in more wasted bandwidth, traffic and latency reaching to almost one minute as shown in Table 3, which has a direct effect on reducing the hit ratio.
7. Summary and Conclusion

Having conducted the previously mentioned experiments, analyzed and made conclusions from the obtained results, implications and recommendations for a Web caching system design are now drawn:

- All factors considered in the evaluation criteria have an important participation in the performance of a Web caching system. The degrees of this participation are shown in Table 4. It is noted that there is no dominating criterion; all have a considerable influence on the performance of a Web caching system.

- The following configurations are recommended:
  - **Client cache size:** Minimum client cache size of 2 GB.
  - **Proxy cache size:** Minimum client cache size of 8 GB.
Abul-Rab and Serhan

- **Cache memory organization**: FAM for both the client's cache and the proxy's cache.

- **Cache fetch policy**: On-demand client cache and pre-fetching at the proxy cache based on matching Web contents are recommended.

- **Cache replacement policy**: LRU client cache replacement policy and LFU for proxy caches.

- The use of homogeneous proxies.

The previous recommendations lead to an optimal system performance of 84% hit ratio, 4.7E+9 bytes wasted bandwidth, 6.8E+9 bytes network traffic and 7 seconds user latency.

**Table 3: The Average Influence of Homogeneous and Heterogeneous Proxies**

<table>
<thead>
<tr>
<th>Factor</th>
<th>Homogeneous</th>
<th>Heterogeneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hit ratio</td>
<td>70</td>
<td>24</td>
</tr>
<tr>
<td>Wasted bandwidth*</td>
<td>6.83</td>
<td>16.7</td>
</tr>
<tr>
<td>Network traffic*</td>
<td>8.83</td>
<td>18.5</td>
</tr>
<tr>
<td>User Latency</td>
<td>24</td>
<td>29</td>
</tr>
</tbody>
</table>

* Measured in e+9 units

**Table 4: Percentage Average of the Influence of Each Evaluation Criterion**

<table>
<thead>
<tr>
<th>Evaluation Criterion</th>
<th>Memory Organization</th>
<th>Fetch Policy</th>
<th>Replacement Policy</th>
<th>Prediction Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hit ratio</td>
<td>23</td>
<td>31</td>
<td>25</td>
<td>21</td>
</tr>
<tr>
<td>Wasted bandwidth</td>
<td>22</td>
<td>32</td>
<td>28</td>
<td>18</td>
</tr>
<tr>
<td>Network traffic</td>
<td>23</td>
<td>27</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>User Latency</td>
<td>21</td>
<td>28</td>
<td>29</td>
<td>22</td>
</tr>
</tbody>
</table>

316
Improving the Effectiveness of Web Caching at the Client's Site

تحسين كفاءة الكاذ في الويب عند طرف التصفح

سامي سرحان و ساميه أبو الرب

ملخص

يتم في هذا البحث تحليل وتقديم الأداء من حيث نسبة الإصابة، عرض النطاق الترددية الضائع، حركة مور الشبكة، وتأخير المستخدم، في حالات ومستويات عمليات مختلفة. يقترح في هذا البحث شكل لفترة ذاكرة كاذ خاص بشبكة الويب العالمية يحقق أداء مثالي تصل فيه نسبة الإصابة إلى 84% وعرض نطاق ترددى ضائع ترددى ضائع (1.7E-9 bytes) وحركة مور على الشبكة (1.6E-9 bytes) و 7 تأخير للفترة المستخدم. يقترح هذا الشكل ملامسة تنظيم ذاكرة كاذ ذات التخطيط كامل الزيت في كل من موقع الموكيل والوكيل. يقترح هذا الشكل ملامسة طريقة أقل استهلاكاً للاستبدال عند موقع الموكيل بطريقة الأقل تأثير استخدام موقع الوكيل.

References

http://vito.in.cs.vt.edu/~chitra/work.html


http://citeseer.nj.nec.com/72530.html

http://citeseer.nj.nec.com/barish99survey.html

http://citeseer.nj.nec.com/barish00world.html


317
   http://www.web-caching.com/mnmtutorial/

     http://citeseer.nj.nec.com/cohen98efficient.html


     http://www.eecs.harvard.edu/~vino/web/usenix96/

[14] Jacobson Q. and Cao P., Potential and limits of Web Prefetching Between Low-

     2000.
     http://citeseer.nj.nec.com/cachedpage/367983/

     Cooperate, 1996.
     http://www.w3.org/Conferences/WWW4/Papers/59/

     http://ltpc45.epfl.ch/WebCaching/presentations/250398/sld001.html


     http://ltpc03.epfl.ch/WebCaching/MoreInfo.htm

     http://www.comp.glam.ac.uk/pages/staff/mreddy/ICPaper/WebNet/


     Prentice Hall, Europe, 1996.

     SIGCOMM, 1996.
     http://topus.cs.vt.edu/~chitra/docs/96sigcomm/

     2000.
     http://citeseer.nj.nec.com/378998.html