Adaptation-Aware Web Caching: Caching in the Future Pervasive Web

Sven Buchholz, Alexander Schill

Dresden University of Technology
Department of Computer Science
D-01062 Dresden, Germany
{buchholz, schill}@rn.inf.tu-dresden.de

Abstract. In the upcoming world of Pervasive Computing, content adaptation is an essential concept to meet the heterogeneous requirements of web users using various web access technologies. However, content adaptation interferes with the effectiveness of web caching. Leveraging the advantages of web caching even in the world of Pervasive Computing is the subject of this paper.
We present an approach that joins the concepts of web caching and content adaptation in a uniform scheme. We have conceived an architecture of hierarchical, independent caching proxies that are aware of the heterogeneous capabilities of the client population. Content adaptation is performed by adaptation services on behalf of the proxies. By this means, the proxies merge their local cache management with the composition of a distributed adaptation path and avoid the interference between content adaptation and web caching.

1 Introduction

In the upcoming world of Pervasive Computing, users access information sources in the World Wide Web by a huge variety of mobile devices featuring heterogeneous capabilities, e.g. with respect to display, data input, computing capacity, etc. These mobile devices coexist with fixed workstations, possibly Internet enabled TV sets, or public information terminals resulting in an even broader diversity of device characteristics and capabilities. The miscellaneous devices are attached to the Internet by a variety of communication systems, such as cellular radio networks, local area wireless networks, dial-up connections, or broadband connections. The various communication systems offer different functionality and heterogeneous characteristics, for example bandwidth, delay, etc.

The key to meet the demands in this heterogeneous environment is the adaptation of the contents to the capabilities of the devices and communication systems. However, content adaptation interferes with effectiveness of web caching that is applied in the World Wide Web to improve performance by avoiding redundant data transfers. Leveraging the advantages of web caching even in the world of Pervasive Computing is the subject of this paper.

The remainder of the paper is organized as follows. In the next section the concept of content adaptation is introduced and different approaches for performing content adaptation are discussed. Section 3 points out the importance of web caching in the application domain of Pervasive Computing and analyzes the interference of content adaptation with the effectiveness of web caching. Our proposed solution, eliminating the interference between content adaptation and web caching, is presented in section 4. Section 5 discusses related work. Finally, some concluding remarks and future directions are given in section 6.

2 Content Adaptation

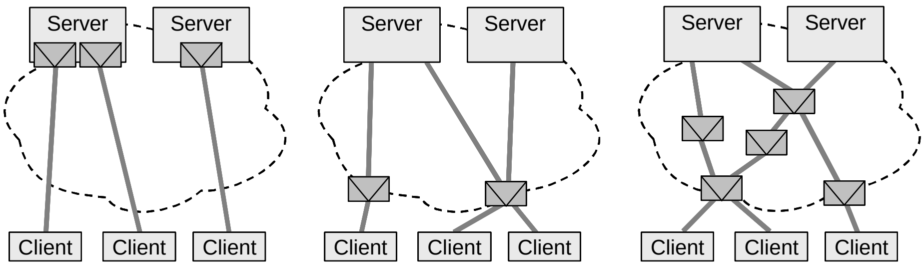
The term content adaptation refers to the modification of the representation of web objects in order to meet the media handling capabilities of the device and the transmission restrictions imposed by the network connection. Such modifications may include: format transcoding (e.g. XML to WML, JPEG to WBMP), scaling of images as well as video and audio streams, media conversion (e.g. text-to-speech), omission or substitution of document parts (e.g. images by a textual representation), or document fragmentation. Even semantic conversions, such as language translation, are considered as content adaptation operations.

A lot of effort has been spent in the field of content adaptation. We distinguish three different approaches for performing content adaptation described in previous work: *server-side adaptation*, *proxy-based adaptation*, and *adaptation paths*.

Server-side adaptation (e.g. [1]) means the servers provide adapted documents (cf. fig. 1a). This can be done either by on-demand dynamic adaptation or by having a repository of pre-adapted documents.

With proxy-based adaptation (e.g. [2]), the documents are provided by the servers in a generic representation[1]. The adaptation is performed on demand by intermediary proxies, which are placed close to the clients (cf. fig. 1b).

The adaptation paths approach (e.g. [3]) is a refinement of the concept of proxy-based adaptation. It is predicated on the finding that adaptation is often composed of



a) Server-side adaptation b) Proxy-based adaptation c) Adaptation paths

Fig. 1. Approaches for Performing Content Adaptation

[1] A generic representation of a text document will typically be some kind of XML document. Images or other multimedia contents may be generically represented by a loss-free compressed format.

elementary, successively applied adaptation steps. Just as with proxy-based adaptation, the server provides the document in a generic representation. On its way to the client, the document passes several distributed adaptation proxies performing the adaptation in a step-by-step manner (cf. fig. 1c). That way a distributed adaptation path is established.

3 Web Caching in the Context of Content Adaptation

The concepts of server-side adaptation or on-demand dynamic adaptation (by a single proxy or an adaptation path) introduced in the previous section have one thing in common: They assume every information access to result in a data flow from the server to the client (if applicable via one or several adaptation proxies). On the other hand, today's Internet providers tend to avoid unnecessary data flows between servers and clients. Therefore, several web caching techniques (cf. [4, 5]) have been developed and deployed. Besides the widely used approach of proxy caching (e.g. [6, 7]), Content Delivery Networks (CDN, cf. [8, 9]) rely on distributed caching of web contents to improve the performance of web accesses.

Web caching exploits sharing effects and temporal locality in client access patterns to satisfy requests by cached copies of the requested objects[2]. By terminating client requests at local caching proxies, the network traffic on the Internet as well as the server load is reduced whereas the response time is improved. Furthermore, the robustness of web services is enhanced as a client may retrieve a document from the proxy cache if the server is unavailable (due to server crash, congestion, or network partitioning). Certainly, significant reductions in response time due to communication costs will be achievable only if there are considerable delays between proxy and server or the bottleneck link is between the proxy and the server. However, this is not always true in Pervasive Computing because often the bottleneck link is the wireless link to the client.

We envision leveraging the advantages of web caching even in the world of Pervasive Computing where content adaptation is necessary. However, the concepts of web caching and content adaptation cannot be considered orthogonal since content adaptation interferes with the effectiveness of web caching. On the other hand, web caching in conjunction with content adaptation may gain even more benefits as adapted contents may be reused, eliminating the cost for re-computing an adapted document. The interferences of web caching and content adaptation are illustrated in the following.

With server-side adaptation servers always serve fully adapted documents. Hence, proxies may cache fully adapted documents only. The drawback of caching fully adapted documents is that sharing benefits are reduced significantly: Though it is likely to have multiple clients requesting the same document, the likelihood of requiring the same representation of the document is considerably smaller.

---

[2] According to [10], 85% of all cache hits are due to sharing whereas 15% account for temporal locality. Nevertheless, in the domain of Pervasive Computing, temporal locality in the users' browsing behavior will result in more locality hits at the proxies because several client devices (such as Smart Phones) do not have a user agent cache.

As opposed to server-side adaptation, proxy-based adaptation allows the caches to be placed between the information sources and the adaptation proxies. Thus the caches store the generic documents that can be adapted to fit all client requests. Whereas this approach fully benefits from sharing, it might undermine the cache memory efficiency. This is due to the fact that generic documents are potentially larger[3] than adapted documents. Accordingly, fewer documents fit in the cache. Cache performance studies [11] suggest that increasing the cache size by a magnitude of 4 may increase the document hit ratio by up to about 25% and the byte hit ratio by up to about 35%. Reducing the size of the documents in the cache by adaptation will gain similar improvements in cache hits. Besides, even higher compression rates than by the magnitude of 4 might be achievable by content adaptation resulting in even higher improvements of the hit ratios. Reduced cache memory efficiency, however, is only an issue in environments with limited cache memory. Experiences with live web proxies teach cache memory is not necessarily a limitative factor today [12]. But the increasing share of streamed multimedia contents will probably turn it into a limitative factor in the future if caches are used for streaming content, too. Besides reduced cache memory efficiency, another problem with proxy-based adaptation in conjunction with web caching is that caching generic documents results in full adaptation costs with every request. The reuse of adapted documents is not possible.

With the adaptation paths approach, caches might be deployed (1) between the server and the first adaptation proxy in the path, (2) between the last adaptation proxy and the client, or (3) between proxies within the adaptation path. The first deployment suffers from reduced sharing benefits as described for server-side adaptation. The second approach may undermine the cache memory efficiency and makes the reuse of adapted contents impossible (cf. proxy-based adaptation). Placing caches within the adaptation path requires coordination between caching and adaptation path composition. However, coordinating caching and content adaptation has not been investigated before. Anyhow, coordination of caching and adaptation path composition may gain optimal benefits from caching. Caches within the adaptation path may always store the representation of an object that is adequately generic to fulfill subsequent requests from heterogeneous clients and sufficiently adapted to be efficient by means of cache memory consumption and adaptation costs.

4 A Unified Approach for Web Caching and Content Adaptation

As seen in the previous section, web proxy caching must be coordinated with content adaptation to allow maximum cache efficiency. In the following we present an approach that achieves the coordination by joining cache management and adaptation path composition in a uniform scheme.

---

[3] Adapted documents are not necessarily smaller in size than generic ones, e.g. if different compressions are used. However, due to the average information content of the generic representation being greater or equal to the one of the adapted representation, adapted documents can be compacted using loss-free compression to be smaller than the (compacted) generic documents.

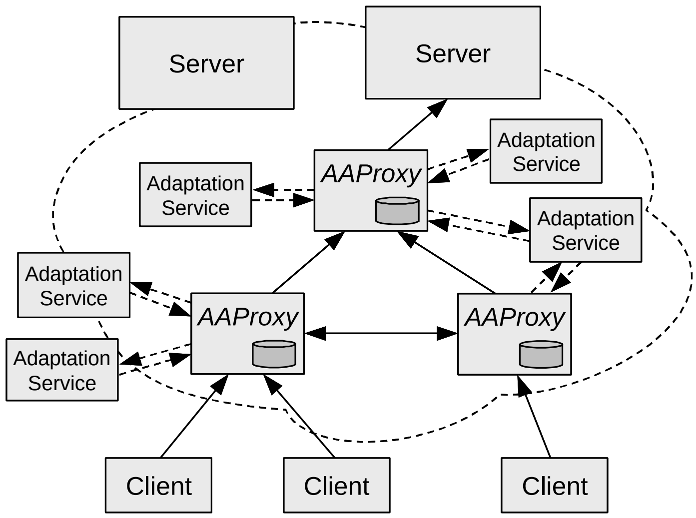


Fig. 2. System Scenario

4.1 System Scenario

Our approach is based on the system scenario presented in figure 2. The architecture consists of a hierarchy of independent caching proxies, having parental and sibling relationships, which cooperate to satisfy client requests. The hierarchy approach is based on the concepts of hierarchical caching, which was first proposed by the Harvest project [6]. In terms of Squid [7], hierarchical caching is the dominant web caching infrastructure on today's Internet.

However, as opposed to Squid and other existing web caching concepts, the caching proxies of the proposed architecture are aware of the heterogeneous capabilities of the clients. The caches store the generic, a partially adapted, or the fully adapted representation of an object depending on which representation seems most appropriate to fulfill subsequent requests. Hence, we call this kind of caching proxy *Adaptation-Aware Proxy* (*AAProxy*).

The content adaptation is performed on behalf of the AAProxies by separate adaptation services (cf. fig. 2). They are invoked by a proxy in a client/server manner to perform particular adaptation operations. The concept of separate adaptation services conforms to the intentions of the ICAP Internet Draft [13].

Certainly, in the application scenario of Pervasive Computing clients should always find their nearest proxy without manual configuration by the user. Therefore we propose the application of transparent caching [14]. Transparent caching means the client addresses its request directly to the web server, but the request gets intercepted at the route and is transparently forwarded to a proxy. By intercepting the request on their route to the server, the requests may always be forwarded to the most appropriate proxy.

Even though the architectural description above relies on the terminology “proxy”, it is not restricted to the classical approach of proxy caching. The functionality of the AAProxies might also be performed by the Edge Servers (cf. [9]) in a Content Delivery Network. In the context of CDN the effectiveness of adaptation-aware caching might be even more significant because technologies such as Edge Side Includes [15] allow for more dynamic contents to be pre-adapted, cached, and reused.

4.2 Request Handling

The basic operation of the proposed system is illustrated in the following (cf. fig. 3). When a user requests an object, the request is forwarded to a nearby AAProxy. Attached to the request the client device provides a description of its media handling capabilities. The media handling capabilities are expressed by means of the media features of the representation the client can handle. We call this representation the *target representation of the client.* As a client may be capable of handling different representations, it may provide a set of multiple target representations. Preferences of particular representations are expressed by assigning different quality values to the different target representations. We propose the application of IETF Media Feature Sets [16] for the purpose of expressing sets of target representations because they allow a flexible yet compact description.

On receiving a request, a AAProxy checks whether the requested object is available from its cache (or possibly from the cache of a sibling proxy) in a representation that meets the requirements of the client or that can be adapted to the client using the available adaptation services. If so, the AAProxy invokes necessary adaptation operations and eventually delivers the object to the client.

In case the request cannot be fulfilled from the cache contents, it is propagated up the hierarchy to the parent AAProxy. Thereby the media handling capabilities description is altered to include those representations that can be adapted to a target representation of the client using available adaptation services. All representations that may be a basis for fulfilling the client request (by invoking zero, one, or multiple adaptation services) constitute the set of *target representations of the proxy*. The adaptation costs and a measure for the quality of the adapted document (taking the client’s preferences for the different target representations into account) are included in the media handling capabilities description. They are used to determine the optimal adaptation path, the path with the best ratio of quality to overall costs.

If the proxy considers the requested object to be popular enough to be cached, target representations that are assumed to fulfill not just the current but even future requests are privileged in the ranking by means of adaptation costs and quality. The ranking is weighted corresponding to the assumed probability that a cached copy can be reused. Assumptions about the probability of reuse are gained from predicting the target representations of future requests based on an evaluation of the media handling capabilities descriptions received with the current and previous requests[4].

---

[4] We have not yet conceived the mechanisms for this evaluation. Nevertheless, we assume the proxies to estimate the capabilities of their client population based on an object spanning interpretation of the media handling capabilities descriptions received with previous requests and to infer the static probabilities for requests for certain target representations.

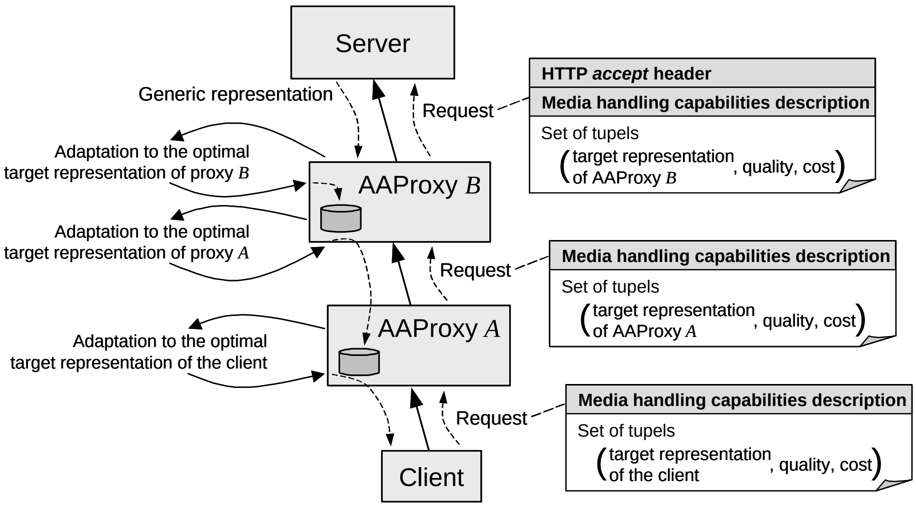


Fig. 3. Request Handling

On receiving a request from a child proxy, the parent AAProxy checks its cache for the requested object. If the request can be fulfilled from the cache, the parent generates its child's target representation that promises the best overall quality-to-costs ratio. Thereby, the costs and quality effects of the adaptations invoked by the parent proxy are also taken into consideration. The target representation promising the best quality-to-costs ratio is called *optimal target representation*.

If the requested object is not available from the cache in an appropriate representation the media handling capabilities description of the request is altered as outlined above and the request is sent to the respective parent. Eventually, the root proxy sends the request to the origin server.

If the origin server understands media handling capabilities descriptions, it determines all target representations of the root proxy that may be fulfilled by a copy of the requested object. The optimal target representation is selected and returned to the root proxy. Thereby the server picks the optimal adaptation path.

Nevertheless, we do not indispensably require the servers to understand media handling capabilities descriptions. Thus, legacy servers can be easily integrated. A legacy server may simply return an object that meets the requirements expressed by the HTTP *accept* header in the request. In this case, the selection of the optimal adaptation path is done by the root proxy. To allow for legacy severs to be integrated we require a root proxy to set the HTTP *accept* header field according to its set of target representations.

While the response travels down the hierarchy, the particular AAProxies invoke adaptation services to make the response meet the optimal target representation of their respective child proxies. By means of successively producing the target representations of the AAProxies in the path, step-by-step adaptation is performed. Moreover, intermediary results of the step-by-step adaptation of cache-worthy objects are stored in the caches of the proxies. Possible other representations of the same object that can be generated (by adaptation) from the new cache copy might be evicted from the cache.

Even the cache replacement mechanisms may exploit the awareness of the AAProxy for content adaptation issues. For instance, instead of completely evicting an object from the cache, the representation may be adapted to consume less memory and still meet the capabilities of the majority of the clients[5]. However, the design of an appropriate cache replacement policy is subject to ongoing research.

5 Related Research

Related research to our work actually spans different research areas. On the one hand this is web proxy caching research. This research area has been investigated for quite a long time and many aspects have been deeply examined [6, 7, 10, 11, 14, 17, 18, 19, 20, 21]. The idea of proxy caching goes back to the CERN proxy [17]. It was further developed in the Harvest project [6], which also created the concepts of hierarchical caching, the basis of our system scenario. The subsequent open source project Squid [7] has further developed the Harvest ideas and integrated inter-cache communication [18, 19] to improve caching efficiency. Even though hierarchical caching by means of Squid is today's most dominant web caching infrastructure, there are several drawbacks that have motivated the development of alternative caching architectures, such as distributed caching [20] or Adaptive Web Caching [21].

One of the weak points of hierarchical caching is that web objects are stored redundantly at different levels in the caching hierarchy. This does not generally apply to our approach as we propose to store different representations (different intermediary results in the adaptation process) at the different levels in the hierarchy. Nevertheless, the algorithms presented in section 4 do not guarantee that there are no redundant copies. The adaptation path considered optimal may absolutely comprise redundant copies.

An alternative concept for improving web performance besides proxy caching is the approach of Content Delivery Networks (CDN; [8, 9]). As opposed to proxy caching, CDNs do not improve the general web access. They speed up the access to selected web sites by caching the contents of the CDN provider's customers. Hence, CDNs are deployed on behalf of the web site providers and not on behalf of the users as with proxy caching. The caching is done close to the users in so-called Edge Servers [9] ran by the CDN provider. According to [8], CDNs may use both pre-caching (which is actually push replication) and just-in-time caching, as applied in proxy caching. Unlike proxy caching, CDNs are not necessarily transparent to the content providers but may use proprietary protocols or protocol extensions. They are used, for instance, to distribute invalidation or update messages to ensure freshness of the cached contents.

---

[5] This is illustrated by the following example. A proxy may have cached a high resolution image that has been exceptionally requested from a full scale laptop computer although the vast majority of the client devices are WAP phones. Such a proxy could free cache memory by scaling-down the image to WAP phone resolution instead of evicting it from the cache.

Recent research in Content Delivery Networks deals with increasing the cacheability of dynamic contents by decomposing complex web objects into fragments with different cacheability. With Edge Side Includes (ESI; [9, 15]), endorsed e.g. by Akamai and Oracle, hypertext documents can be described as a composition of a template and multiple document fragments. The resulting hypertext document is assembled at the Edge Servers of the CDN. Hence, the fragments may be cached and updated independent from each other.

As mentioned before, the approach presented in this paper may be applied in CDNs. Technologies such as ESI complement our approach to provide for more dynamic contents to be pre-adapted, cached, and reused. The assembly of the result document can be considered an adaptation step in the adaptation path.

In addition to caching architectures, cache replacement and admittance policies, cache coherency, deployment options, and implementation details have been subject to research. An overview of research issues in web caching is given in [4, 5].

Content adaptation research is another area with relevance to our approach. Early efforts date back to the *DeleGate* Gopher proxy [22] for Kanji transcoding that started in 1994. Leveraging content adaptation to adapt to the capabilities of mobile devices was investigated by the *Daedalus* project at the UC Berkeley. It has proven that on-demand dynamic adaptation of text and images by proxies is feasible and powerful [2]. The concept of proxy-based dynamic adaptation was further refined by the UC Berkeley’s Ninja Project [3]. The project proposes a robust, scalable architecture for web access by heterogeneous devices. In the Ninja project, on-demand dynamic adaptation is done in a step-by-step manner by several adaptation proxies establishing an adaptation path, the so-called Ninja Path. The ideas of Ninja Paths are the basis for the step-by-step adaptation used in our approach. Ninja assumes a centralized instance of the path subsystem performing the path composition and implementation [23]. While the Ninja approach guarantees the composition of valid paths, path optimization is not addressed.

Algorithms for composing and optimizing a distributed adaptation path have been dealt with by [24, 25]. Ooi et al. [24] describe an algorithm to distribute the computation of multimedia streams across multiple multimedia gateways. However, the algorithm presented in the paper optimizes the computation with respect to resource consumption only. Aspects such as sharing of intermediary results are not addressed. Likewise, caching is not an issue. Kasim et al. [25] present a more general and more abstract approach. They describe algorithms to determine a valid distributed adaptation path to make a multimedia object in a certain representation available at the client node. The origin object may reside on different nodes in different representations. In [25], the authors describe bottom-up as well as top-down algorithms. Furthermore, they have proven the top-down algorithm to determine the optimal adaptation path. These algorithms, however, do not deal with caching or sharing of intermediary adaptation results. Even though, the bottom-up algorithm is the basis for the algorithm we are developing for the proposed architecture.

A lot of effort has been spent on the adaptation of web documents. Automatic re-authoring of HTML pages to adapt them to the capabilities of mobile devices is done by the *Digestor* system [26] and its successor, the m-link system [27], by Fuji Xerox Palo Alto Laboratory (FXPAL). Those systems use heuristics to understand web documents to enable content adaptation. However, the potentials of such heuristics are limited. In order to overcome those limitations, other work considers generic XML-based representations of documents that are augmented by meta information to allow smarter adaptation (e.g. [28]). The approach presented in this paper does not rely on the one or the other concept for the adaptation of web contents. It is meant to deal with arbitrary means of content adaptation.

In the context of streaming media, caching of quality adaptive streams has been investigated [29, 30]. This is related to our approach as streams may be cached in different representations (qualities). However, as opposed to our ideas they only take the adaptation of multi-media streams by dropping layers or frames into account. Rejaie et al. [29] consider every layer of a multi-layered encoded stream as a separate object by means of the cache replacement algorithm and apply a weighted *LFU-Aging* algorithm to determine a *victim layer*, which is to be evicted from the cache. Yu et al. [30] do not only take multi-media streams but even non-continuous media, such as text and images, into account. They present a media-characteristic-weighted replacement policy that considers single frames of media streams as separate objects just as non-continuous media objects. Media-characteristic-weighted replacement means that dependencies between different objects (e.g. between I, P and B frames of an MPEG stream) are taken into account and different media types may be assigned different priorities. Adaptation, however, is limited to dropping particular frames.

As opposed to our approach, those approaches for proxy caching of quality adaptive multimedia streams only consider simple one-dimensional adaptations (dropping frames or layers of a multi-media stream). Our approach, however, considers arbitrary adaptations as far as they can be performed by an adaptation service. Furthermore, they do not exploit caching hierarchies to benefit from different representation of objects at different hierarchy levels. Accordingly, only our approach describes the negotiation of the optimal target representations for the different proxies and thereby the composition of a distributed adaptation path.

6 Conclusion

This paper discussed the application of web caching in the domain of Pervasive Computing where content adaptation is necessary to adjust to the particularities of the heterogeneous client devices. We pointed out that content adaptation interferes with the effectiveness of web caching. That is why novel schemes that coordinate web caching and content adaptation are necessary.

We presented an approach that joins the concepts of web caching and content adaptation in a uniform scheme. We conceived hierarchical, independent caching proxies that are aware of the heterogeneous capabilities of the client population. Content adaptation is performed by adaptation services on behalf of the proxies. That way, the proxies merge their local cache management with the composition of a distributed adaptation path.

The presented approach is still subject to ongoing research. Currently we are investigating the algorithms for negotiating the adaptation path in detail. Open questions include the estimation of adaptation costs and quality effects of operations that enable free scaling of objects. Furthermore, we want to refine the definition of caching costs, as the current algorithm does not take the costs for re-requesting evicted objects into account. Thereby we will also design a media features aware replacement policy for the proxy caches. Moreover, early experiences with our algorithm suggest the proxies might have to maintain a lot of information about the client population and will possibly exchange a significant amount of extra data for target representations. If this occurs we will need heuristics to reduce the overhead.

The benefit of the approach presented in the paper shall be evaluated using simulation. Therefore we are working on a flexibly parametrizable trace synthesizer to generate traces that estimate the access patterns of future pervasive web users. First intermediary results have been published in [12].

References

1. Oracle 9i Application Server: Wireless Edition, Technical White Paper, Oracle Corporation, Redwood City, CA, USA (2000).
2. Fox A., Gribble, S.D., Chawathe, Y., Brewer, E.A.: Adapting to Network and Client Variation Using Active Proxies: Lessons and Perspectives. In: A Special Issue of IEEE Personal Communications on Adaptation (1998).
3. Gribble, S.D., Welsh, M., von Behren, R., Brewer, E.A., Culler, D., Borisov, N., Czerwinski, S., Gummadi, R., Hill, J., Joseph, A., Katz, R.H., Mao, Z.M., Ross, S., Zhao, B.: The Ninja architecture for robust Internet-scale systems and services. In: Computer Networks 35(4) (2001).
4. Wang, J.: A survey of web caching schemes for the Internet. In: ACM Computer Communication Review 29(5) (1999).
5. Barish, G., Obraczka, K.: World Wide Web Caching: Trends and Techniques. In: IEEE Communications, Internet Technology Series (2000).
6. Chankhunthod, A., Danzig, P.B., Neerdaels, C., Schwartz, M.F., Worrell, K.J.: A Hierarchical Internet Object Cache. In: Proceedings of the USENIX Technical Conference, San Diego, CA, USA (1996).
7. Wessels, D., Claffy, K.: ICP and the Squid Web Cache. In: IEEE Journal on Selected Areas in Communication 16(3) (1998).
8. Liste, M.: Content Delivery Networks (CDNs) - A Reference Guide. In: Cisco World, White Papers (2001). URL: http://www.ciscoworldmagazine.com/webpapers/2001/03_thrupoint.shtml [2002-10-15]
9. Turbo-Charging Dynamic Web Sites with Akamai EdgeSuite. Akamai White Paper AKAMWP-TCD1201, Akamai Technologies, Inc. Cambridge, MA, USA (2001). URL: http://www.akamai.com/en/resources/pdf/Turbocharging_WP.pdf [2002-10-15].
10. Duska, B., Marwood, D., Freeley, M.J.: The Measured Access Characteristics of World-Wide-Web Client Proxy Caches. In: Proceedings of the USENIX Symposium on Internet Technologies and Systems. Monterey, CA (1997).
11. Mahanti, A.: Web Proxy Workload Characterisation And Modelling. M.Sc. Thesis, Department of Computer Science, University of Saskatchewan (1999).
12. Buchholz, S., Jaensch, S., Schill, A.: Flexible Web Traffic Modeling for New Application Domains. In: Proc. of the IASTED International Conference on Applied Modelling and Simulation (AMS 2002), Cambridge, MA, USA (2002).
13. Elson, J., Cerpa, A.: ICAP the Internet Content Adaptation Protocol. Internet Draft, The ICAP Protocol Group (2001). URL: http://www.i-cap.org/spec/icap_specification.txt [2002-10-17]

14. Cohen, A., Rangarajan, S., Singh, N.: Supporting Transparent Caching with Standard Proxy Caches. In: Proceedings of the 4th International Web Caching Workshop, San Diego (1999).
15. Tsimelzon, M., Weihl, B., Jacobs, L.: ESI Language Specification 1.0. Akamai Technologies, Inc. Cambridge, MA, USA , Oracle Corporation, Redwood City, CA, USA (2001). URL: http://www.esi.org/language_spec_1-0.html [2002-10-16]
16. Klyne, G.: A Syntax for Describing Media Feature Sets. RFC 2533 (1999).
17. Luotonen, A., Altis, K.: World-Wide Web Proxies. In: Computer Networks and ISDN Systems 27(2), Elsevier Science (1994).
18. Wessels, D., Claffy, K.: Internet Cache Protocol (ICP), version 2, RFC 2186 (1997).
19. Rousskov, A., Wessels, D.: Cache Digest. In: Proceedings of the 3rd International WWW Caching Workshop, Manchester, UK (1998).
20. Tewari, R. Dahlin, M., Vin, H.M., Kay, J.S.: Beyond Hierarchies: Design Considerations for Distributed Caching on the Internet. Technical Report TR98-04, University of Texas at Austin (1998).
21. Zhang, L., Floyd, S. Jacobson, V.: Adaptive Web Caching. In: Proceedings of the NLANR Web Cache Workshop, Boulder, CO (1997).
22. Sato, Y.: DeleGate Server. (1994). URL: http://www.delegate.org/y.sato/DeleGate/ [2002-10-17]
23. Chandrasekaran, S., Madden, S., Ionescu, M.: Ninja Paths: An Architecture for Composing Services over Wide Area Networks. CS262 class project writeup, UC Berkeley (2000). URL: http://ninja.cs.berkeley.edu/dist/papers/path.ps.gz [2002-10-17]
24. Ooi, W.; van Renesse, R.: Distributing Media Transformation Over Multiple Media Gateways. In: Proc. Of the 9th ACM International Multimedia Conference, Ottawa, Canada (2001).
25. Candan, K., Subrahmanian, V., Rangan, P.: Collaborative Multimedia Systems: Synthesis of Media Objects. In: IEEE Transactions on Knowledge and Data Engineering 10(3) (1998).
26. Bickmore, T., Girgensohn, A., Sullivan, J.W.: Web Page Filtering and Re-Authoring for Mobile Users. In: The Computer Journal 42 (6) (1999).
27. Schilit, B.N., Trevor, J., Hilbert, D.M., Koh, T.K.: m-Links: An Infrastructure for Very Small Internet Devices. In: Proc. of the 7th Annual Int'l Conference on Mobile Computing and Networking, Rome, Italy (2001).
28. Goebel, S., Buchholz, S., Ziegert, T., Schill, A.: Device Independent Representation of Web-based Dialogs and Contents. In: Proceedings of the IEEE Youth Forum in Computer Science and Engineering (YUFORIC'01), Valencia, Spain (2001).
29. Rejaie, R., Yu, H., Handley, M., Estrin, D.: Multimedia Proxy Caching Mechanism for Quality Adaptive Streaming Applications in the Internet. In: Proc. of IEEE INFOCOM 2000, Tel-Aviv, Israel (2000).
30. Yu, F., Zhang, Q., Zhu, W., Zhang, Y.: Network-Adaptive Cache Management Schemes for Mixed Media. In: Proc. of the 2nd IEEE Pacific-Rim Conference on Multimedia (IEEE-PCM), Beijing (2001).