To appear in the Proceedings of the 9th International Conference on User Modeling, Johnstown, PA, 2003

Performance Evaluation of User Modeling Servers Under Real-World Workload Conditions

Alfred Kobsa[1], Josef Fink[2]

[1] School of Information and Computer Science, University of California, U.S.A.
kobsa@uci.edu
[2] Department of Mathematics and Computer Science, University of Essen, Germany
Josef-Fridolin.Fink@t-online.de

Abstract. Before user modeling servers can be deployed to real-world application environments with potentially millions of users, their runtime behavior must be experimentally verified under realistic workload conditions to ascertain their satisfactory performance in the target domain. This paper discusses performance experiments which systematically vary the number of profiles available in the user modeling server, and the frequency of page requests that simulated users submit to a hypothetical personalized website. The parameters of this simulation are based on empirical web usage research. For small to medium sized test scenarios, the processing time for a representative mix of user modeling operations was found to only degressively increase with the frequency of page requests. The distribution of the user modeling server across a network of computers additionally accelerated those operations that are amenable to parallel execution. A large-scale test with several million active user profiles and a page request rate that is representative of major websites confirmed that the user modeling performance of our server will not impose a significant overhead for a personalized website. It also corroborated our earlier finding that directories provide a superior foundation for user modeling servers than traditionally used data bases and knowledge bases.

1 Introduction

Before user modeling (UM) servers [1, 2] can be deployed to real-world application scenarios with potentially millions of users, their runtime behavior must be experimentally tested under realistic workload conditions to ascertain their satisfactory performance in the target environment. The parameters of such experiments, and specifically the workload of simulated user interactions that cause requests to the UM server, should thereby closely resemble the target domain. The few existing performance studies of UM servers [3, 4] and of directory servers [5, 6] however all employed synthetic workloads that are not based on empirical results about web usage behavior.

Unfortunately, most existing web traffic data are not very useful for empirically based workload experiments since they are based on proxy logs (e.g., [7, 8]) or web server logs (e.g., [9, 10]). Such data has limited value since it does not reflect all communication that would ordinarily take place between browsers and web servers [11].

For instance, browsers may connect to web servers via several proxies, and numerous caches may affect the amount of traffic between browsers and web servers. Most published studies are moreover based on websites of research institutions, which are not very representative for users' typical website visits[1] and presumably also not for the navigation behavior that is exhibited at more typical sites [10].

2 Web Usage Patterns

Rozanski et al. [13] recently conducted a comprehensive analysis of click-stream data collected by the audience measurement service Nielsen//NetRatings. The data was collected at the *client side* from a panel of 2,466 Internet users over several months. In a first step, the researchers identified 186,797 user sessions[2]. Subsequently, they tested a variety of session characteristics with regard to their suitability for clustering these sessions. The most differentiating session characteristics were the following:

Session length: defined as the length of a single user session on the Internet.

Time per page: denotes the time interval between two subsequent web page requests.

Category concentration: the percentage of time a user stays at websites of the same category (e.g., news, sports, entertainment, real estate).

Site familiarity: the percentage of time a user stays at familiar sites, i.e. sites she had previously visited four or more times.

Based on these characteristics, Rozanski et al. carried out a cluster analysis and distinguished the following patterns of web usage (in parentheses their relative frequency):

Quickie sessions (8%): These are short (one minute) visits to one or two familiar sites, to extract specific bits of information (e.g., stock quotes, sports results). Users visit 2.2 pages per site on average, and spend about 15 seconds on a page.

Just the Facts sessions (15%): Here users seek and evaluate specific pieces of information at related sites (e.g., compare product offers). Sessions last 9 minutes on average. Users visit 10.5 sites and 1.7 pages per site, with about 30 sec. per page.

Single Mission sessions (7%): Users focus on gathering specific information or completing concrete tasks (e.g., finding the website of a scientific conference and registering for it). They visit two websites on average, which belong to the same category (e.g., search engines or portals). Users quite carefully read the content of (frequently unfamiliar) web pages in approximately 90 seconds. The average session length is 10 minutes, and 3.3 pages per site are being visited.

Do It Again sessions (14%): These are focused on sites with which the user is familiar (e.g., online banks, chat rooms). Users spend about two minutes for each page. The average session lasts 14 minutes, with 2.1 sites and 3.3 pages per site being visited.

Loitering sessions (16%): Users visit familiar "sticky" sites, such as news, gaming, telecommunications/ISP, and entertainment. Sessions last 33 minutes, with 8.5 sites

[1] E.g., [12] found that 35% of users' surfing time is spent on merely 50 (commercial) sites.

[2] A session represents the total time from when a user signs on to the Internet to when she signs off, or to the point when her activity ceases for more than an hour.

and 1.9 pages per site being visited (two minutes per page on average).

Information Please sessions (17%): Users gather broad information from a range of often unfamiliar websites from several categories (e.g., they collect facts about a specific car model, find a dealer, negotiate a trade-in, and arrange a loan). Users visit 19.7 websites and 1.9 pages per site. The average session length is 37 minutes, and pages are viewed for one minute on average.

Surfing sessions (23%): They appear random, with users visiting nearly 45 sites in 70 minutes on average (about one minute per page and 1.6 pages per site).

Over time, users can engage in several, if not all, session types, depending on how different their tasks are. Rozanski et al. found, e.g., that two-third engaged in five or more session types and 44 percent in all seven session types.

3 Workload Simulation

Our user modeling server comprises the following components:

Directory Component, which stores assumptions about the user in terms taken from a domain taxonomy. It utilizes the iPlanet (Sun ONE [14]) LDAP Directory Server.

User Learning Component (ULC), which learns about the user (specifically about her interests) through univariate significance analysis of her usage characteristics.

Mentor Learning Component (MLC), which learns about a user's interests via alike "mentors" found through Spearman correlation of their usage characteristics.

Domain Inference Component (DIC), which uses rules for inferences about the user.

The details of these components are not relevant for the purposes of this paper. We refer the reader to [16]. [15] additionally discusses a prototype application in mobile computing, and [17] a deployment to a major news site in Germany.

To test the performance of our UM server under different workload conditions, we simulated users' interaction with a hypothetical personalized website. Each user thereby follows one of the abovementioned session types. The content of each web page is characterized by 1-3 terms taken from the domain taxonomy. Web page requests by a user lead to add and query operations in his user profile on the UM server: the terms of the requested web page are processed and added to his interest model, and the user's interests in terms of the domain taxonomy are queried to personalize a web page that was requested by him. As a shortcut though, we omit the web server in our simulation and represent web pages by their characteristic terms only.

Our first experiment for small to medium sized personalized applications was a two-factor design with the following parameters:

- N (number of existing profiles in the UM server): 100, 500, 2,500, or 12,500[3].
- W (number of web page requests per second): 0.5, 1, 2, or 4[4].

[3] The corresponding user population is larger since only some users opt for personalization (5% in Yahoo and 25% in an early version of myAltaVista [18], 64% in Excite [19]).

[4] Based on data from [20], one can estimate that three of four German websites receive less than four page requests per second on average.

For every factor combination, we generate a test plan with N user profiles. The behavior of currently active users of the hypothetical website is simulated by clients of our user modeling server. Clients are divided into seven classes, which represent the aforementioned session types. A class *i* comprises c_i clients which exhibit the web page request behavior that is characteristic for their class. The c_i clients of a class *i* create a total workload of w_i page requests per second. The combined workload of all clients equals the preset frequency of page requests W (i.e., 0.5, 1, 2, or 4 pages per second). We assume that w_i / W approximates the observed type frequency of class *i* (this assumption is corroborated by a manual count of the frequencies of Quickie and Just the Facts sessions at several German websites, such as [17]). Table 1 shows the test plan for a workload of 2 pages per second.

Table 1: Simulation environment for 2 page requests per second (* = figure rounded)

	Session type characteristics		Test bed parameters	
Variables Session types	Relative type frequency	Interval between requests	Requests/sec. (w_i)*	No. of clients (c_i)*
Quickies	8%	15 sec	0.13	2
Just the Facts	15%	30 sec	0.30	9
Single Mission	7%	90 sec	0.14	13
Do It Again	14%	120 sec	0.28	34
Loitering	16%	120 sec	0.33	39
Information, Please	17%	60 sec	0.35	21
Surfing	23%	60 sec	0.47	28
Total	100%		2.00	146

We assume Zipf-like distributions of the frequencies in which

1. terms from the domain taxonomy become characteristic terms for web pages;
2. users engage in a new session with our hypothetical website;
3. web pages are requested by users ("page popularity").

Assumption (1) is based on the fact that term frequency distributions in documents tend to follow Zipf's law [21]. (2) is an estimate based on several studies regarding the frequency and duration of people's Internet usage (e.g., [22]). (3) is derived from the observation that web page popularity follows a Zipf-like distribution $1/i^{\alpha}$, where *i* is the popularity rank of the web page and α an adjustment for the server environment and the domain. [10, 23-25] recommend different values for α. We followed [10] who analyzed the MSNBC news site since their study was the most recent and their site the most similar to our own target site. The authors recommend an α between 1.4 and 1.6, hence we opted for α=1.5 and use this value for all three distributions.

We assume further that our UM server has to process the following operations for personalizing a requested web page[5]:

- Three search operations with Zipf-distributed terms from the domain taxonomy, namely for personalizing the page header (e.g., user-tailored banner ads), the navigation section (e.g., personalized links), and the content part (e.g., personalized news). We assume one exact and two substring searches.
- One add operation for communicating the 1-3 characteristic terms of a web page as an interest event to the UM server.

For implementing our simulation environment, we took advantage of Directory Mark 1.1, a benchmark suite for LDAP servers from Mindcraft [26]. Directory Mark simulates clients that simultaneously access an LDAP server and reports a variety of performance data. For each test scenario, we generated an appropriate number of user profiles as well as transaction scripts that implement the workloads for each of the session types introduced earlier. To avoid starting a test run with all user profiles being empty, we introduced a warm-up phase during which the profiles became initially populated (lasting 10 minutes for 100 user profiles, 50 minutes for 500 profiles, etc.). During our tests, we collected and recorded 269 measures for the UM server and its components. Major results will be described below.

4 Small to Medium Scale Application Scenario

Our first series of experiments was carried out with a hardware configuration that would be typical for small web stores or news sites. In one test variant, all user modeling functionality resided on a single platform. In a second variant, we distributed the four components of our UM server across a network of four computers. In both conditions, a PC with an 800 MHz CPU, 512 MB of RAM and a 100 Mbps network card hosted the environment that simulated users submitting page requests.

4.1 Single Platform Tests

In the single platform tests, the complete UM server (i.e., Directory Component, ULC, MLC, and DIC) was running on a single PC with two 800 MHz processors, 1 GB of RAM, a RAID controller with two 18.3 GB UW-SCSI hard disks, and a 100 Mbps network card. The software used was Windows NT 4.0, iPlanet Directory Server 4.13 and VisiBroker 3.4. The learning and inference components were compiled with Java 1.2.2 and used the Java Hot Spot Server Virtual Machine 2.0.

Fig. 1 shows the mean times that our UM server takes to perform the four user model operations for personalizing a page from the viewpoint of our hypothetical web application. The results for all 16 value combinations of our independent variables are charted. In general, mean times increase only degressively with the number of page

[5] Note that many personalized websites do not provide personalization on all pages, which lowers the load of the UM server.

requests and user profiles. In two cases (namely for 100 and 500 profiles), the times for four page requests per second are even lower than for two. This advantageous response time behavior is mainly due to database caching in the LDAP directory server. The more user model operations are being sent to the server for a given number of user profiles, the faster this cache gets filled and the more operations can therefore be directly served from cache memory. We also see that all mean times for 12,500 users are higher than those for smaller numbers of user profiles, while the mean times for 100, 500, and 2,500 user profiles appear quite similar (except for 2,500 users and four pages). We assume that this effect results mainly from a higher hit rate (i.e., probability that a specific piece of information is contained in cache memory) in those cases that have a smaller number of user profiles.

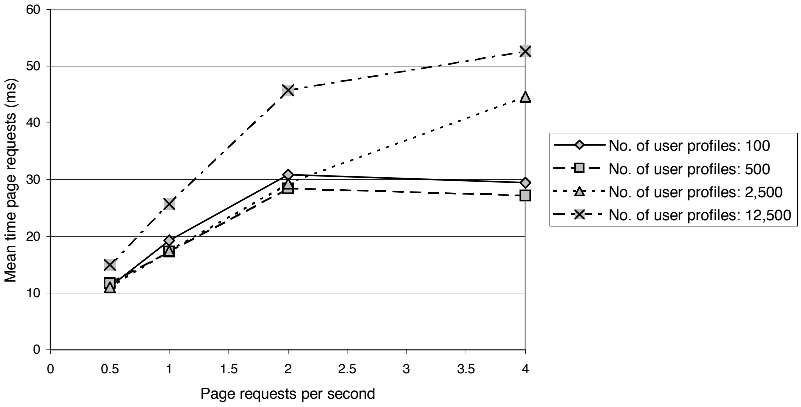

Fig. 1: Mean processing times for personalizing a web page

The overall performance and scalability of our UM server appears highly satisfactory. Even in the case of four page requests per second and 12,500 user models, the mean time to execute four user model operations and to return the results to 288 clients in parallel is smaller than 53 ms. The 99% confidence interval for the means does not exceed ± 0.24 ms due to the large sample size. The mean times plus one / two standard deviations never exceed 78 / 103 ms. A more detailed analysis shows that this graceful performance degradation occurs for both add and search operations. Since the overhead caused by the UM server is minor, web-based applications will be able to provide personalized services while responding within the desirable limit of one second and, in any case, the mandatory limit of ten seconds [27]. The moderate surge of the mean response time when the number of clients and user profiles increases does not suggest impending performance cliffs and scalability limits.

4.2 Multiple Platform Tests

In the multi-platform scenario, only the Directory Component was running on the mentioned dual processor computer. The three other components of the UM server were each installed on a separate 600-800 MHz single processor PC with 100 Mbps network card. Fig. 2 compares several measurements for both scenarios. We see that

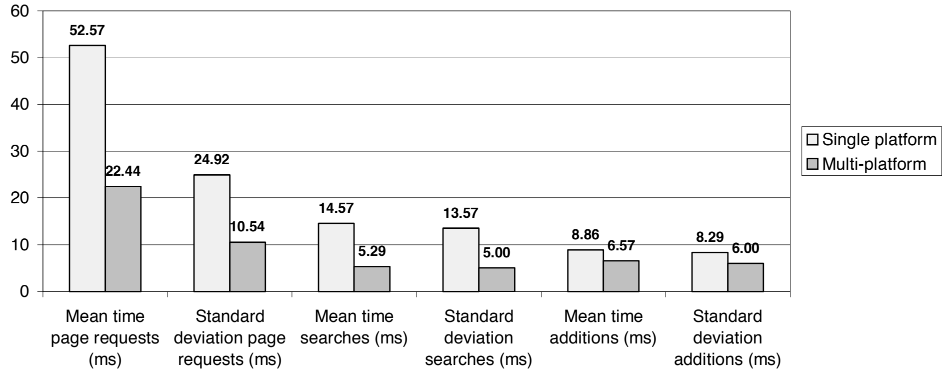

Fig. 2. Single-platform vs. multi-platform performance (12,500 profiles, 4 pages/sec)

the mean time for processing the four user model operations that personalize a web page plunges to 22.44 from 52.57 milliseconds, and its standard deviation to 10.54 from 24.92 milliseconds (i.e., nearly 60% in both cases). The single most important reason for this improvement is the considerably better search performance. The mean search time falls to 5.29 from 14.57 ms (-64%), and its σ to 5 from 13.57 ms (-63%). Less impressive is the performance gain of add operations: the mean time drops to 6.57 from 8.86 ms (-26%), and σ to 6 from 8.29 ms (-28%).

The distribution of our UM server across a network of four computers improved its performance considerably. Search operations benefit most from the relieved dual processor computer, since they can now be carried out concurrently by the directory server. Add operations with their inherent need for multi-user synchronization [16] can take less advantage of the additional hardware resources.

4.3 Evaluation of the Learning Components

So far, we discussed the performance of our UM server from the viewpoint of our hypothetical web application. Now we turn to the individual components of our server: the statistics-based User Learning Component, the similarity-based Mentor Learning Component, and the rule-based Domain Inference Component. These components operate concurrently to the Directory Component. Fig. 3 shows the mean processing times of the ULC and the MLC for the single platform scenario. The performance of the DIC (which is comparable to that of the ULC) is discussed in [16].

For the ULC, mean times seem to mainly depend on the number of user profiles. They grow degressively with increasing page requests, which is mainly due to the queue-based architecture of the ULC (it allows for bulk processing of submitted events and for interim storage of interest probabilities in the main memory, thereby saving costly updates of the user profile). All recorded mean times are smaller than four seconds, which is highly satisfactory since it permits keeping track of users' changing interests even between consecutive page requests. The ULC fully supports this inter-request learning for all session types and workloads we tested.

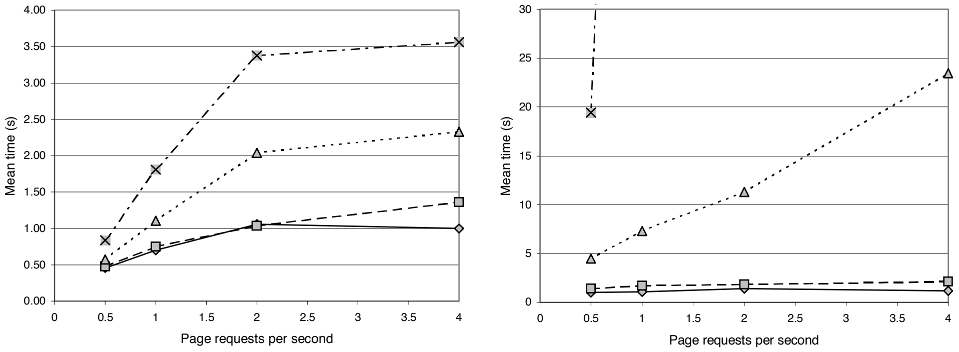

Fig. 3 Mean processing times of statistics and mentor based learning components (see Fig. 1 for legend)

The performance of the MLC is less good. For 100, 500 and 2,500 profiles, all means are below 24 seconds but grow progressively with increased page request rate. Except for Quickies, this still allows for a prediction of user interests and preferences between consecutive page requests. The response time deteriorates considerably though for 12,500 user profiles: 19 sec. for 0.5 and 141 sec. for 1 page/sec, but more than 2 hours for 2 and 4 pages/sec. In the latter two cases, the MLC presumably cannot keep pace with the stream of user arrivals and approaches its performance limits.

5 Large Scale Application Scenario

The successful simulation results for a small to medium sized user-adaptive website put us in the position to run a series of experiments on a much larger scale. The most notable one comprised eight million user profiles[6] and a workload of approximately 42 web page requests per second[7]. To realize this workload, we employed a total of 1,794 simultaneous clients in several testbeds. The UM server was installed on a Fire V880 from Sun's entry-level server segment [30] under Solaris 8, with eight 750 MHz processors, 8 MB cache per processor, 32 GB of RAM, and more than 200 GB of disk space. To take full advantage of the available hardware, we increased the cache size of the Directory Component and each learning component to 2 GB. The user modeling server was implemented in version 5.1 of iPlanet Directory Server. Otherwise the design of this experiment was very comparable to the one described in Section 3.1.

The results were again very encouraging. Our UM server showed a mean response time of 35 ms for personalizing a web page (i.e., for performing three LDAP search and one add operations). This user modeling performance should easily allow a personalized application to stay well below the desirable response time limit of one second and, in any case, below the mandatory limit of ten seconds. None of the several million search and add operations that were submitted by our simulated users

[6] MSN had about 8 and AOL 34 million subscribers at the end of July 2002 [28].

[7] This workload roughly equals that of the largest German news portal with nearly 15 million unique users [29], which is about 15-20% the size of the top three U.S. portals.

failed or timed out. Overall, the quality of service offered by our server seems highly satisfactory.

Another lesson from simulating the user modeling workload of real-world application environments was in terms of hardware sizing. The sizing characteristics of our server closely resemble those reported in the literature for its Directory Component. For example, [31] mentions the following rules of thumb for the number of CPUs that are necessary to process LDAP operations: "With Directory Server 4.0, search performance will scale almost linearly with the addition of up to 4 CPUs. In this range, you can expect to see 500-1,000 queries per second for each CPU. Beyond 4 CPUs, the resulting increase in performance per CPU is less but still significant".

The resource needs of the learning and inference components of our UM server depend on the number of these components (each can be present or absent, and instantiated multiple times), and several parameters that determine, e.g., the learning frequency, the size of the correlation space, etc. As far as the allocation of processor resources is concerned, we found that an even distribution between the Directory Component, and the learning and inference components, seems to be a good solution.

In conclusion, we regard our empirically founded approach of simulating the UM workload of real-world application environments as highly promising. It allows us to experimentally verify and predict the deployment characteristics of a UM server under various workload conditions. Our experience with actual installations of our UM server in commercial environments confirmed that this approach and the developed simulation testbed are an indispensable tool for real-world personalization. It also corroborated the findings in [15] (which were based on theoretical considerations and work by others) that directories provide a superior foundation for UM servers than traditionally used data bases and knowledge bases.

References

1. Kobsa, A.: Generic User Modeling Systems. User Modeling and User-Adapted Interaction **11**, (2001) 49-63. http://www.ics.uci.edu/~kobsa/papers/2001-UMUAI-kobsa.pdf
2. Fink, J. and Kobsa, A.: A Review and Analysis of Commercial User Modeling Servers for Personalization on the World Wide Web. User Modeling and User-Adapted Interaction **10**, (2000) 209-249. http://www.ics.uci.edu/~kobsa/papers/2000-UMUAI-kobsa.pdf
3. VanderMeer, D., Dutta, K., and Datta, A.: Enabling Scalable Online Personalization on the Web. 2nd ACM Conf on Electronic Commerce, Minneapolis, MN (2000) 185-196.
4. Datta, A., Dutta, K., VanderMeer, D., Ramamritham, K., Navathe, S. B.: An Architecture to Support Scalable Online Personalization on the Web. VLDB Journal **10** (2001) 104-17.
5. Keung, S. and Abbot, S.: LDAP Server Performance Report. (1998), http://www.bnelson.com/sizing/docl/ldapsPerformance.html
6. Wang, X., Schulzrinne, H., Kandlur, D., and Verma, D.: Measurement and Analysis of LDAP Performance. ACM SIGMETRICS Conference, Santa Clara, CA (2000) 156-165.
7. Duska, B. M., Marwood, D., and Feeley, M. J.: The Measured Access Characteristics of World-Wide-Web Client Proxy Caches. USENIX Symposium on Internet Technologies and Systems, Monterey, CA (1997) 23-35.

8. Gribble, S. D. and Brewer, E. A.: System Design Issues for Internet Middleware Services: Deductions from a Large Client Trace. USENIX Symposium on Internet Technologies and Systems, Monterey, CA (1997).
9. Almeida, V., Bestavros, A., Crovella, M., and Oliveira, A.: Characterizing Reference Locality in the WWW. Fourth International Conference on Parallel and Distributed Information Systems (1996) 92-103.
10. Padmanabhan, V. and Qiu, L.: The Content and Access Dynamics of a Busy Web Site: Findings and Implications. ACM SIGCOMM (2000) 111-123.
11. Fenstermacher, K. D. and Ginsburg, M.: Mining Client-Side Activity for Personalization. Fourth Workshop on Advanced Issues in Electronic Commerce and Web Information Systems (WECWIS), Newport Beach, CA (2002) 44-51.
12. 35 Percent of Surfing Time is Spent on 50 Sites. Computer Scope Ltd. (1999), http://www.nua.com/surveys/index.cgi?f=VS&art_id=905355323&rel=true
13. Rozanski, H., Bollman, G., and Lipman, M.: Seize the Occasion: Usage-based Segmentation for Internet Marketers. Booz-Allen & Hamilton, Inc. (2001), http://www.strategy-business.com/media/pdf/03-20-01_eInsight.pdf
14. Sun ONE Directory Server. Sun Microsystems (2002), http://wwws.sun.com/software/products/directory_srvr/home_directory.html
15. Fink, J. and Kobsa, A.: User Modeling in Personalized City Tours. Artificial Intelligence Review **18**, (2002) 33-74. http://www.ics.uci.edu/~kobsa/papers/2002-AIR-kobsa.pdf
16. Fink, J.: User Modeling Servers: Requirements, Design, and Evaluation. Department of Mathematics and Computer Science: University of Essen, Germany (2003).
17. Fink, J., Koenemann, J., Noller, S., and Schwab, I.: Putting Personalization into Practice. Communications of the ACM **45**, (2002) 41-42.
18. AltaVista Announcement. Compaq (1999), http://www.compaq.com/newsroom/presspaq/012699/schrock.html
19. Excite Network Online Media Kit. Excite (2002), http://www.excitenetwork.com/ advertising/index/id/Directmarket|ListRental|3|1.html
20. Online Usage Data Nov. 2001 (in German). IVW (2001), www.ivw.de/data/index.html
21. Zipf, G. K.: Human Behavior and the Principle of Least Effort. Reading: Addison (1949).
22. Patrick, A. and Black, A.: Implications of Access Methods and Frequency of Use for the National Capital Freenet. (1996) debra.dgbt.doc.ca/services-research/survey/connections/
23. Glassman, S.: A Caching Relay for the World Wide Web. First International Conference on the World-Wide Web, Geneva, Switzerland (1994).
24. Zipf Curves and Website Popularity. (1997), http://www.useit.com/alertbox/zipf.html
25. Breslau, L., Cao, P., Fan, L., Phillips, G., and Shenker, S.: Web Caching and Zipf-Like Distributions: Evidence and Implications. INFOCOM'99 (1999) 126-134.
26. DirectoryMark: The LDAP Server Benchmarking Tool. Mindcraft (2002), http://www.mindcraft.com/directorymark/index.html
27. Nielsen, J.: Usability Engineering. San Diego, CA: Academic Press (1993).
28. MSN Hits 300 Million Unique Monthly Users. Jupitermedia (2002), http://cyberatlas.internet.com/big_picture/traffic_patterns/article/0,,5931_1457661,00.html
29. Top 10 Web Properties for the Month of October 2002. Netratings Inc. (2002), http://epm.netratings.com/de/web/NRpublicreports.toppropertiesmonthly.html
30. Entry-Level Servers. Sun Microsystems (2002), http://www.sun.com/servers/entry/
31. Nelson, B.: Sizing Guide for Netscape Directory Server. (2002), http://www.bnelson.com/sizing/doc2/Directory4_0-SizingGuide.html