Web Caching Evaluation from Wikipedia Request Statistics

Gerhard Hasslinger
Deutsche Telekom
Darmstadt, Germany
gerhard.hasslinger@telekom.de

Mahmoud Kunbaz
Chemnitz Univ. of Tech., Germany
Chair of Communication Networks
mahmoud.kunbaz@s2014.tu-chemnitz.de

Frank Hasslinger
Darmstadt Univ. of Tech.
Darmstadt, Germany
frank.hasslinger@stud.tu-darmstadt.de

Thomas Bauschert
Chemnitz Univ. of Tech., Germany
Chair of Communication Networks
thomas.bauschert@etit.tu-chemnitz.de

***Abstract* — Wikipedia is one of the most popular information platforms on the Internet. The user access pattern to Wikipedia pages depends on their relevance in the current worldwide social discourse. We use publically available statistics about the top-1000 most popular pages on each day to estimate the efficiency of caches for support of the platform. While the data volumes are moderate, the main goal of Wikipedia caches is to reduce access times for page views and edits. We study the impact of most popular pages on the achievable cache hit rate in comparison to Zipf request distributions and we include daily dynamics in popularity.**

***Keywords* — Wikipedia daily top-1000 statistics, Zipf distributed requests, web caching strategies, hit rate, simulation**

I. Introduction

A. Content delivery and caching systems on the web

The steadily ongoing growth of traffic and improvements in the performance for web services rely on an infrastructure of distributed server and caching systems, which are integrated into content delivery networks (CDNs) and cloud architectures [4][10][17][19][20][22]. Caching enables shorter transport paths of the data to the users, which also means shorter delays and higher availability when a source can be chosen within distributed and replicated content delivery systems. Large caching architectures for support of a global user population often include several hierarchical levels of caches combining core data center locations with smaller edge caching sites.

B. Statistics about the Wikipedia web site and user requests

Wikipedia is a well known web sites which offers a large collection of articles as a steadily updated online dictionary. Wikipedia includes over 40 million pages, making 7473 volumes of 700 pages in an online ebook version or 23 TByte of storage according to the Size_of_Wikipedia page [25] already in 2014. The total number of Wikipedia requests per day is in the range of 600 - 750 million with peak rates going beyond 50 000 request per second.

In this work, we analyze statistical data with regard to the caching efficiency. Wikipedia is supported by the MediaWiki distributed cache server architecture from at least four main locations [25]. While the data and traffic volume is smaller than for global video and IPTV services [4][24], low delay and fast page loading times are a main concern for Wikipedia because the quality of experience is an important asset to make a web site attractive for the users. In 2015, a speedup of the median Wikipedia page load time from 1.2-1.3s down to 0.8-0.9s is reported on Wikimedia page Global_traffic_routing [25]. Therefore preloading of first paint overview information was introduced before the complete page content is transmitted. Moreover, in peak periods of high user request rates, cache servers help to avoid congestion. Otherwise, users are redirected to an error page "404.php", which is among the top-10 most "requested" pages in December 2016.

C. Efficient cache support for content delivery on the web

Our focus is not especially on the Wikimedia cache servers, but more general on conclusions that we can draw from the Wikipedia request statistics on the cache hit rate and on caching polices for optimizing the hit rate. Whether reduced delay, traffic load or other objectives are in the focus, the hit rate is a common basic measure of the cache performance. Hit rates depend on the user access pattern, where a concentration of requests on a small set of popular pages supports caching performance.

The size of a cache and the policy for inserting and replacing objects in the cache have main impact on the hit rate [3] [18]. Based on top-1000 statistics, we can model caching behavior only with limited size. However, small size caches are relevant and allow conclusions also for similar performance characteristics of larger caches. There is a general tradeoff between the size and the forwarding speed of caching systems. Large TByte caches on cheap SSD storage for the mass of data are usually augmented by several cache levels with much faster and smaller DRAM and SRAM memory, which can process the data at high line rates. Thus caches of different type and size are relevant in high performance CDN/cloud servers [21].

Largely different sizes of web objects were also relevant in the past, but seem less important today, since large web objects are subdivided into data chunks and cache storage is steadily growing. Thus we assume a unique size of cached objects, corresponding to data chunks.

We use simulation for evaluating caching efficiency, because analytical results on cache hit rates are available only in a few special cases [5][8][12].

We start the main part in Section II by adapting Zipf distributions to the top-1000 Wikipedia requests. In Section III, we study the dynamics in page popularity visible through daily changes. Section IV briefly summarizes web caching strategies, whose performance is evaluated in Section V with regard to Wikipedia request pattern, followed by the conclusions.

II. Daily Wikipedia top-1000 statistics & Zipf requests

A. Evaluations of Wikipedia Top-K Request Statistics

For studying the performance of caching systems under realistic access pattern of a popular web platform, we refer to the

978-3-901882-90-6/17 ©2017 IFIP

daily top-1000 page request statistics made available by Wikimedia since August 2015 [25]. All our evaluations are for English Wikipedia pages on <en.wikipedia.org> [14]. Most relevant characteristics for the evaluation of caching from such data are

- the distribution of the frequencies of requests of the top-K most popular pages, and
- the rate of change in the popularity of objects from a day to the next one.

Figure 1 shows the number of requests to the top-K most popular English Wikipedia pages for K = 1, 10, 100, 1000. A considerable variability is visible, which is higher for smaller sets of top-K objects. Requests to the top page vary by a factor of almost 40 between 185 000 - 7 million per day, whereas requests to the top-1000 are in the range 16 - 58 million. Note, that we excluded about a dozen often requested meta pages from all evaluations including the Wikipedia main page, several search pages and the error page “404.php”. Those pages attract partly extreme request peaks, which do not fit into Zipf request pattern. On the other hand, including those meta pages would lead to higher but also more variable hit rates.

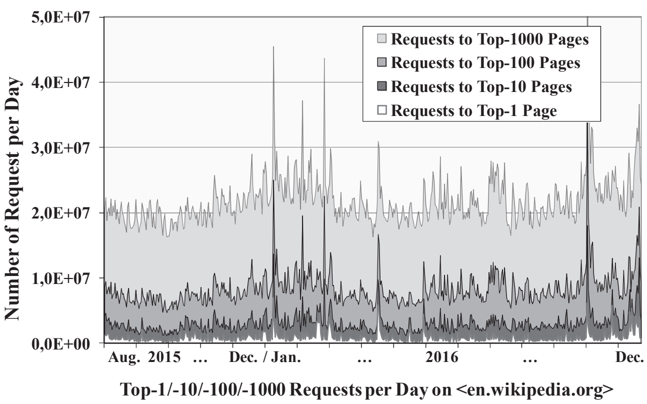

Figure 1: Daily Wikipedia Request Statistics

B. Adaptation of Zipf Distributions to the Request Statistics

In a first step, we check whether the daily top-1000 request distributions are in the shape of a Zipf distribution. Many studies have confirmed Zipf’s law as an appropriate model for access pattern to content on the Internet. They include video platforms like YouTube [2], IP-TV channel selection [4][19], P2P file sharing systems [23], web shops etc. As a consequence, a small set of popular web objects attracts most user requests and thus makes small caches efficient.

We consider Zipf distributions of finite support for a content catalogue of N objects with request probabilities $z(r)$ being determined for the objects’ popularity ranks $r \in \{1, 2, \ldots, N\}$:

$$z(r) = \alpha r^{\beta} \ \text{ with } \beta < 0; \quad \alpha = z(1) = 1/\sum_{r=1}^{N} r^{\beta} > 0; \qquad (1)$$

where β is an adaptive shape parameter and α a normalization constant. Access probabilities are becoming more unbalanced for $\beta \to -1$. Many case studies based on different sets of web request measurement traces [2][8][16] report a good fit of Zipf adaptations within the paramter range $-0.5 > \beta > -1$.

Figure 2 compares Zipf distributions (dotted line curves) to a set of usual daily Wikipedia page request distributions. The figure gives an impression of typical deviations over the ranks of the top-1000 requests and illustrates how the shape of Zipf distributions is influenced by the parameter β.

Let $W_{\mathrm{CDF}}(d,k) = R_d(k)/R_d(1000)$ denote the cumulative request distribution function (CDF), where $R_d(k)$ is the sum of requests to the top-k pages (k = 1, ..., 1000) at a day d = 1, ..., 519 in the period from Aug. 2015 - Dec. 2016. We determine the deviation $\Delta_{W\leftrightarrow Z}(d,k)$ of $W_{\mathrm{CDF}}(d,k)$ and a corresponding Zipf adaptation $Z_{\mathrm{CDF}}(d,k) = \Sigma_{r \le k}\, z_d(r)$ for a day d by the difference in the ranks, i.e., the distance of a pair of curves in the direction of the rank axis in Figure 2:

$$\Delta_{W\leftrightarrow Z}(d,k) = j_k - k \ \text{ where } \ Z_{\mathrm{CDF}}(d,j_k) \le W_{\mathrm{CDF}}(k) < Z_{\mathrm{CDF}}(d,j_k+1).$$

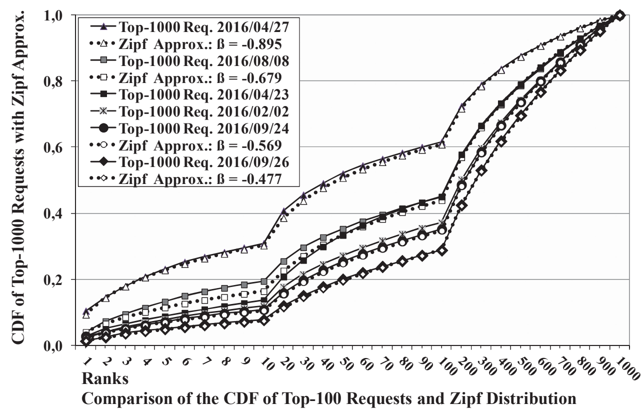

Figure 2: Zipf adaptations to daily Wikipedia requests

We also obtain the mean and maximum absolute rank deviation

$$\Delta^{\mathrm{mean}}_{W\leftrightarrow Z}(d) = \frac{\sum_k \left|\Delta_{W\leftrightarrow Z}(d,k)\right|}{1000}; \quad \Delta^{\max}_{W\leftrightarrow Z}(d) = \max_{k=1,\ldots,1000} \Delta_{W\leftrightarrow Z}(d,k). \qquad (2)$$

The parameter β of the Zipf adaptation is determined per day in order to minimize $\Delta^{\mathrm{mean}}_{W\leftrightarrow Z}(d)$, where we could find a unique, minimizing β for all days without having a proof of this property. Figure 3 shows the mean and maximum deviations of eq. (2) and the parameter β in adaptations for each of the 519 days in the considered period. We found Zipf adaptations in the ranges $-0.477 > \beta > -0.895$, $0.19 < \Delta^{\mathrm{mean}}_{W\leftrightarrow Z}(d) < 10.7$ and $1 \le \Delta^{\max}_{W\leftrightarrow Z}(d) \le 31$ The mean rank deviation over all ranks and days is 3.2.

Figure 3 also shows how the precision of Zipf adaptations varies in terms of the rank deviation over the 519 days, and indicates the daily values of β. On most of the days, the maximum absolute rank deviations are below 10, where 5 out of 519 days show a perfect match with maximum deviations $\Delta^{\max}_{W\leftrightarrow Z}(d)$ of ±1.

On the other hand, Figure 4 illustrates limitations of Zipf adaptations for the worst case example with $\Delta^{\mathrm{mean}}_{W\leftrightarrow Z}(d) \approx 10.69$. In this case, a single parameter adaptation is insufficient to cover the whole range. Instead, three adaptations are shown for fitting one of the three ranges 1-10, 11-100 and 101-1000. While the adaptations make a good fit in one of the sub-ranges, they largely deviate in the other sub-ranges.

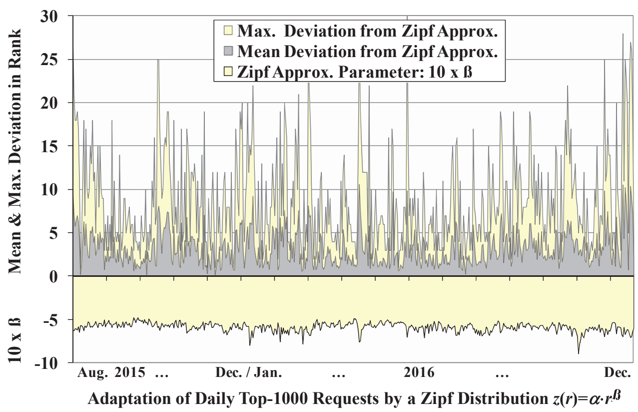

Figure 3: Zipf adaptations to daily Wikipedia page requests

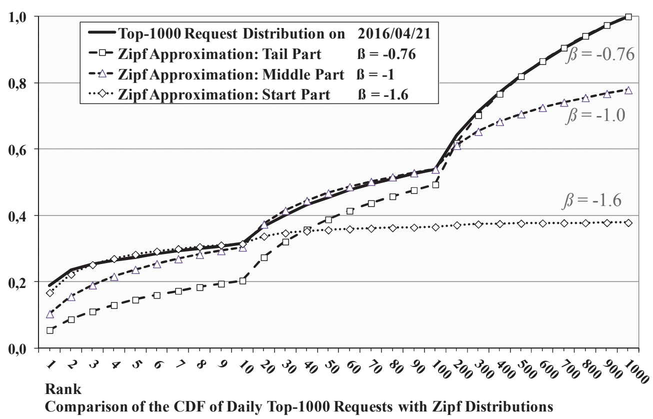

Figure 4: Worst case Zipf adaptation for Wikipedia requests

When we consider the number of requests to top-K Wikipedia pages over longer time periods of a week, a month etc. then the variability in the request distributions is smoothing. Corresponding Zipf approximations show small or moderate deviations. The mean values of requests to the top-1000 pages over all days from Aug. 2015 - Dec. 2016 can be approximated by a Zipf distribution with β = -0.595, with mean and maximum absolute rank deviations of 3.06 and 7, respectively.

III. Dynamics in Daily Top-K Wikipedia Page Requests

The concentration of requests on a small set of top-K pages as expressed in Zipf-like distributions is one main criterion for caching efficiency. Moreover, the dynamics in the popularity of the pages is also relevant, where high dynamics and a frequent occurrence of new or previously unpopular pages in the top-1000 statistics requires more frequent reloading and makes caches less efficient [3][9][17][20][24].

As an indicator of popularity dynamics, we determine how many pages are newly entering the top-K ranks each day, which were not among the top-K on the previous day. Figure 5 shows the maximum, minimum and the mean number of new pages per day that appear on the 2nd, 3rd, …, 519th day in the Wikipedia statistics. E.g., in case K = 100, 42.4% of the top-100 pages are replaced on the next day with 36.7% of requests addressing the new pages per day. Both fractions are decreasing for $K \to 1000$ towards 27.5% new top-1000 pages attracting 23.7% of requests per day. The maximum and minimum values indicate high variability in the daily dynamics, where extremely high dynamics is visible on only a few days which hold the maximum above 40% over the range K = 1, …, 1000.

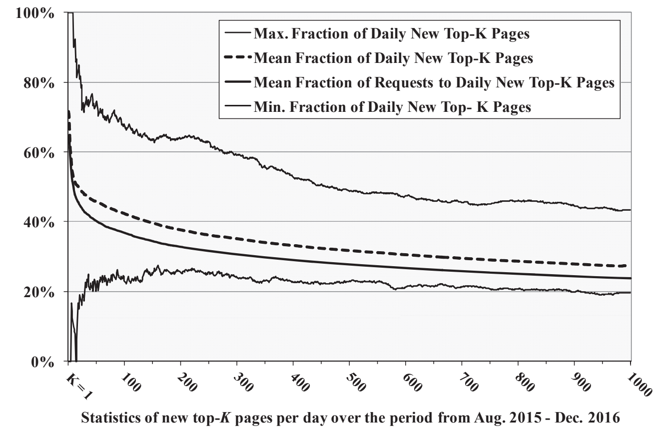

Figure 5: Dynamics in Wikipedia pages and their requests

Dynamics in request pattern can be studied on longer time scales (weeks, months …), but the daily changes are most relevant for caching efficiency. Considering single pages, we can distinguish the impact of one-timers, i.e. pages that appear once in the top-1000 of a day from constantly present pages. Almost half of the pages that appear in the top-1000 in the 17 months period are seen only once, but they get only 4.2% of all requests in the statistics. The 70 pages that are always present attract 8.7% of the requests. More details are shown in Table 1. Consequently, it seems worthwhile to store some of the long term present pages even in small caches, but the evaluation in the next section shows that the least frequently used (LFU) strategy of caching the most often requested pages is far below optimum, at least for the complete long term statistics.

Table 1: Statistics for pages that appear x-times in the daily top-1000 from Aug. 2015 – Dec. 2016

Number of Days in the Top-1000	Number of Pages	% of Pages	% of Requests to those Pages
519 (on each day)	70	0.15	8.7%
400 - 518	131	0.27	12.7%
300 - 399	111	0.23	6.6%
200 - 299	197	0.41	8.3%
100 - 199	445	0.92	12.7%
11 - 99	5251	10.9	30.9%
2 - 10	19772	41.0	15.8%
1	22281	46.1	4.2%

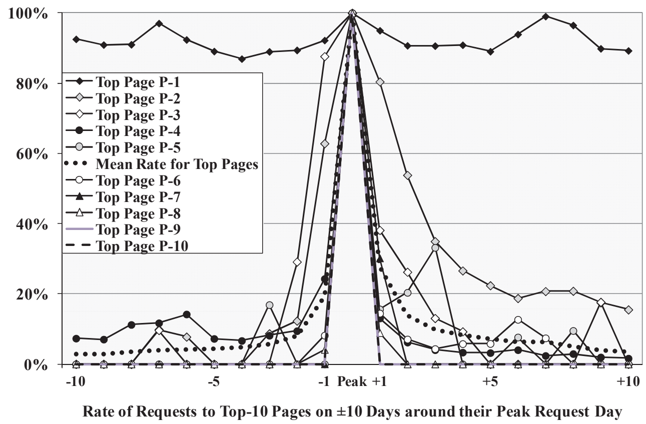

Figure 6: Requests to top pages on days around peak popularity

Finally, an evaluation of Wikipedia top-1000 pages illustrates the change of popularity in the top-10 pages, i.e. the 10 pages that attract most requests on a day. We assign the day of their peak popularity in the middle of Figure 6 and consider the ratio of requests relative to the peak on ±10 days around the peak as a percentage of the peak request rate.

It becomes visible that peak request rates of single Wikipedia pages seldom hold on over several days, but are mostly spontaneous. We have sorted all 1830 top-10 pages appearing on the 519 days due to the sum of percentages on ±10 days around the peak and picked 10 examples P-1, …, P-10 equally spaced over the sorted range. The mean percentages per day for all top-10 pages are included as dotted line curve.

There are only a few pages with long term high request rate like P-1. Most top pages show a request level of 10% - 80% of the peak rate on a few days around the peak. However, 819 out of the 1830 top-10 pages are not among the top-1000 on the day before their peak. Therefore predictions for preloading of pages with rising popularity into the cache are subject to high uncertainty at least on a daily basis, while some yearly correlation due to anniversaries can only partly help. The request fluctuations of Wikipedia pages are completely different from behavior of top objects on video and P2P platforms, which are observed to show a steep rise to peak popularity followed by a slow decrease over time [6][7][23].

IV. Performance of Web Caching Efficiency based on Wikipedia Statistics

We extend previous evaluation methods [11][12] of the hit rate for different caching strategies, which assume independent (IRM) Zipf distributed requests, to include measurement based request pattern derived from the daily Wikipedia statistics.

A. *Web Caching Strategies: LRU, SG-LRU, LFU*

The least recently used (LRU) principle provides a basic and usual caching strategy with a simple cache update scheme. LRU puts the currently requested object on top of a cache implemented as a stack and replaces the bottom object with the longest time span since the last request, if the cache is full.

However, LRU doesn't provide flexibility to manage the cache content based on object specific properties, which may include transport costs, delays, availability or other preferences from the content providers', ISPs' or users' perspective [1][10][22]. Therefore alternative caching strategies have been proposed in literature [13][15][16][18], which improve LRU cache hit rates, but are more complex and not widely adopted.

In recent work we contributed in this field by proposing Score Gated LRU (SG-LRU) caching schemes, which combine the simple LRU implementation with a score rating per each object, giving full flexibility to adapt the caching strategy to arbitrary cost and benefit criteria by setting corresponding object scores [11][12]. We include scores as an additional criterion to admit objects to an LRU cache. LRU always puts the currently requested object on top of the cache, whereas SG-LRU admits a new object to the cache only if it's score is higher than the score of the cache bottom object. Otherwise, the bottom object is put on top, simply by shifting the cache top pointer to this object in a double linked list implementation. We experienced that the SG-LRU variant is sufficient to collect objects with highest scores in the cache, even if it takes longer than maintaining a sorted list of objects according to scores, which requires high update effort [12].

Our evaluations of caching strategies also include the least frequently used (LFU) principle as a reference, which counts previous requests per object and prefers most often requested objects in the cache. If a request counter per object is used as score function then SG-LRU behaves similar to LFU. In the sequel, we study SG-LRU with a request count over a sliding window of the last W requests, where the window size W can be adapted as a memory back log. SG-LRU with sliding window scores resembles LRU in the special case of window size $W = 1$ and behaves similar to LFU for sufficiently large window size. In this way, SG-LRU covers the "LRFU spectrum" [15] of cache strategies with limited memory "between" LRU and LFU, but we stay with the simple LRU update scheme instead of more expensive sorting methods, e.g. heap sort [15].

V. Evaluation of Cache Efficiency

A. *Cache simulation under daily changing request pattern*

We extend a cache simulator developed in [11][12] for daily changing requests due to measurements of top-1000 Wikipedia requests per day in the time from Aug. 2015 - Dec. 2016. On each day, we assume constant request probabilities $R_d^{(k)}/R_d$ to the pages, where $R_d^{(k)}$ is the number of requests to the page on rank k in the top-1000 statistics on day d and R_d denotes the total number of top-1000 requests on that day. We simulate the cache load on each day by performing R_d independent requests to the top-1000 pages of the day with request probabilities $R_d^{(k)}/R_d$, i.e. we assume daily changing independent request models $\mathrm{IRM}_1, \ldots, \mathrm{IRM}_{519}$. On a new day, the cache starts with the content of the end of the previous day. It is left to the (SG-)LRU caching strategy to reoptimize the cache content for the request distribution on the next day.

In order to control the precision of simulated cache hit rates, we use the second order statistics, i.e. the variance of the computed hit rates over several times scales, i.e. for request sequences of different length [11]. The estimated standard deviation of the hit rate is below 0.0002 for the simulation of a daily workload with at least 16 million top-1000 requests. It is even smaller for the mean value over the complete period, confirming sufficient significance for hit rate results on 3 digits.

Figure 7 shows hit rates for all days in the 17 months lasting period for SG-LRU caches with a size of M = 100 pages and for different window sizes with regard to past requests. The lowest curve is for the pure LRU cache hit rate or SG-LRU with W = 1. In this case, our simulations on each day fit to the Che approximation [5][8] with mean relative deviation of less than 1‰. Although this result again confirms the Che approximation to be "remarkably accurate" [8], and despite of the mathematical explanation and asymptotic properties derived in [8], no bounds are known on its accuracy. Therefore simulations seem indispensible to obtain LRU hit rates including a concrete estimate of their accuracy within confidence intervals.

Figure 7 includes curves on SG-LRU strategies with a count of past requests over windows of size W=400, W=4000 and W=40 000, respectively. The mean SG-LRU hit rates over all 519 days are shown in Table 2. The case W=40 000 almost coincides with to the upper bound given by

$$h_{\max} = \sum_{d=1}^{591} \sum_{k=1}^{M} R_d^{(k)} \Big/ \sum_{d=1}^{591} R_d \qquad (3)$$

where $\Sigma_{k=1}^{M} R_d^{(k)} / R_d$ is the upper bound of the cache hit rate for IRM$_d$ requests on a single day, i.e. by putting the top-M most requested Wikipedia pages into the cache for that day, and $h_{\max}$ computes a mean value over all requests in the considered time period.

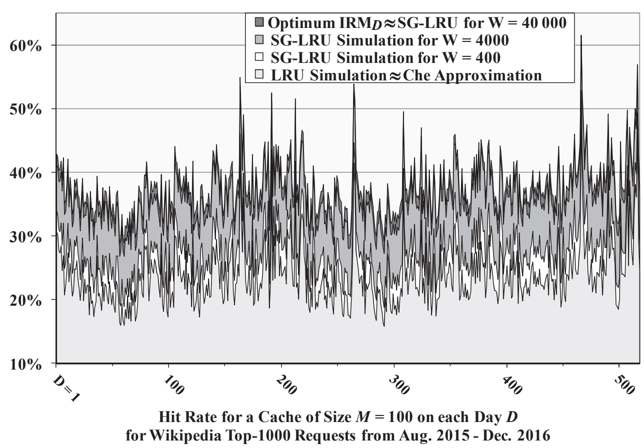

Figure 7: Cache hit rate based on Wikipedia request statistics

Table 2: Mean cache hit rates over 519 days

	LRU	SG-LRU: Window Size W =			Upper bound (3)
		400	4 000	40 000	
Hit Rate	24.4%	28.5%	36.6%	38.2%	38.4%

We can only take daily changes into account, without information on dynamics and correlation of requests during a day. Correlated requests on shorter time frames can lead to higher caching efficiency, but the dynamics on time frames of days is experienced to be more important in caching studies of measurement traces [24]. Implementations by cache providers usually combine continuous updates per request with a complete daily update of the whole cache during low traffic hours at night, including prefeteching of new popular objects based on prediction and/or available information about request pattern changes for the next day [17][20].

Finally it is remarkable that Figure 7 shows peaks on several days with cache hit rates about 1.5-fold above the mean in all curves. Those peaks in the hit rate coincide with peaks in the total number of top-1000 as well as top-10 requests in Figure 1 on the same days. This indicates a favourable property of caches to increase their efficiency with spontaneous peaks in the request load.

B. Impact of Cache Size on Cache Hit Rates

The cache size is a key factor for the achievable hit rate. In Table 3, the LFU hit rate stands for keeping the M most requested pages constantly in the cache over the almost 1.5 year period. LRU is better, but also stays essentially below the optimum hit rate according to eq. (3), which is almost fully exploited by SG-LRU with an appropriate window size W.

Table 3: Cache hit rate depending on the cache size

Cache Size M	**10**	**20**	**50**	**100**	**200**
LFU Hit Rate	4.1%	6.1%	10.7%	16.2%	24.3%
LRU Hit Rate	4.4%	7.6%	15.1%	24.4%	38.2%
SG-LRU Hit Rate	14.0%	19.3%	28.5%	38.2%	50.3%

C. Evaluation for regional caches with lower request load

The latter results for daily changing request distributions in Table 3 and Figure 7 are still close to the bound of eq. (3). Actually, the mean number $R_d \approx 2.2 \cdot 10^7$ of requests per day goes far beyond the phase for adapting the cache content to new request pattern of a day. We extend the simulation results by distributing the request load equally over C caches, each of which serves $\lceil R_d / C \rceil$ independent requests per day. Thus, we run the same simulation as for the results in Figure 7**,** but only for $\lceil R_d / C \rceil$ instead of R_d requests per day.

With decreasing number of requests per day per cache, the adaptation phase to new popular pages becomes more relevant. In case C = 20 000 we have $\lceil R_d / C \rceil \approx 1100$ daily requests per cache, while 27.5% of top-1000 web pages are changing in the same time span. Higher dynamics in the popularity of pages makes the request pattern less predictable and, as a consequence, reduces the cache hit rate.

The results for SG-LRU in Figure 8 start at the daily LFU hit rate level of eq. (3) for small C. This level is hold until $C \approx$ 200, i.e. as long as there are more than 100 000 requests per day, before hit rates essentially decline for larger C. The curves are obtained for an optimized window size W as the only parameter of the SG-LRU strategy. A request count statistics for

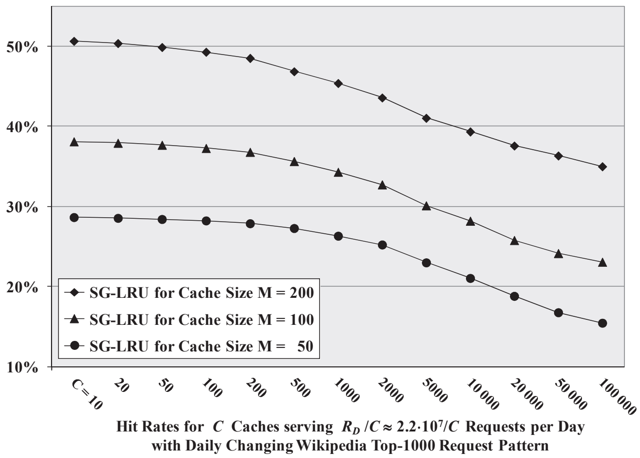

Figure 8: Cache hit rate for daily changing request pattern

one day corresponds to $W = \lceil R_d / C \rceil$, which indicates a reasonable first estimate for W. We run simulations with several values of W approaching the optimum, which is in the order of one million for $C = 10$, going down a few hundred for $C = 10^6$. With decreasing W, SG-LRU is developing towards LRU. We also have to run simulations at least $\lceil C/100 \rceil$-fold in order to get results within appropriate confidence intervals [11].

Conclusions and Outlook

Our main evaluation results of Wikipedia's daily top-1000 request statistics show

- that the Zipf law of eq. (1) provides a fairly good match of most daily top-1000 request distributions,
- that largest deviations from Zipf laws coincide with days showing high peak request rates, which also lead to higher cache hit rates than for the basically favourable Zipf case,
- that small caches of 1% of the content catalogue already show considerable cache performance, see Table 3,
- that neither LFU nor LRU can exploit the cache efficiently, whereas SG-LRU variants including request counts exploit the upper bound of eq. (3) for daily changing IRM requests,
- that Wikipedia request pattern is dynamic with about 27.5% of new pages appearing in each day's top-1000, but such dynamics is negligible for the cache hit rate, if a load of 100 000 or more requests is served per day.

Naturally, this work based on daily top-1000 statistics ignores the larger portion of requests to pages beyond the top-1000, thus restricting conclusions to small caches for the most popular pages. We aim at studying cache performance with complete request measurement traces on other web platforms to gain broader experience on how web cache performance depends on usual request pattern for web content.

Acknowledgements

We would like to thank the operators of the Wikimedia platform for their effort of providing daily updated statistical data about requests to Wikipedia. Our work has received funding from the European Union´s Horizon 2020 innovation programme in the SSICLOPS project under grant agreement No. 644866 <www.ssiclops.eu>. This work reflects only the authors' views and the European Commission is not responsible for any use that may be made of the information it contains.

References

[1] A. Araldo, D. Rossi, F. Martignon, Cost-aware caching: Caching more (costly items) for less (ISPs operational expenditures), IEEE Transactions on Parallel and Distributed Systems (TPDS), 27 (2016) 1316-1330

[2] L. Breslau et al., Web caching and Zipf-like distributions: Evidence and implications, Proc. IEEE Infocom (1999)

[3] M. Busari and C. Williamson, On the sensitivity of Web proxy cache performance to workload characteristics, IEEE Infocom (2001) 779-794

[4] M. Cha et al., Watching TV over an IP network, Proc. 8th SIGCOMM Conf. on Internet Measurement (IMC), Athens, Greece (2008) 71-84

[5] H. Che, Y. Tung, and Z. Wang, Hierarchic web caching systems: modeling, design and experimental results, IEEE JSAC 20(7) (2002) 1305-14

[6] B. Cohen, Incentives build robustness in BitTorrent (2003) 1-5

[7] F. Figueiredo et al., TrendLearner: Early prediction of popularity trends of user generated content (2014) <http://arxiv.org/abs/1402.2351>

[8] C. Fricker, P. Robert and J. Roberts, A versatile and accurate approximation for LRU cache performance, IEEE Proc. 24th International Teletraffic Congress, Kraków, Poland (2012)

[9] M. Garetto, E. Leonardi and S. Traverso, Efficient analysis of caching strategies under dynamic content popularity, Proc. IEEE Infocom (2012)

[10] G. Hasslinger and F. Hartleb, Content delivery and caching from a network provider's perspective, Special Issue on Internet based Content Delivery, Computer Networks 55 (Dec. 2011) 3991-4006

[11] G. Hasslinger, K. Ntougias and F. Hasslinger, Performance and Precision of Web Caching Simulations Including a Random Generator for Zipf Request Pattern, Proc. 18th MMB Conf., Münster, Germany, Springer LNCS 9629 (2016) 60-76

[12] G. Hasslinger, K. Ntougias, F. Hasslinger and O. Hohlfeld: Performance Evaluation for New Web Caching Strategies Combining LRU with Score Based Object Selection, Proc. International Teletraffic Congress ITC-28, Würzburg, Germany (2016) 321-330

[13] M. Hefeeda, O. Saleh, Traffic modeling and proportional partial caching for P2P systems, IEEE/ACM Trans. on Networking 16/6 (2008) 1447-60

[14] M. Kunbaz, Dynamics in the Popularity of User Request on IP Platforms and their Impact on Caching Efficiency, Research Project Report, TU Chemnitz (2016) 1-54

[15] D. Lee et al., LRFU: A spectrum of policies that subsumes the least recently used and least frequently used policies, IEEE Transactions on Computers 50/12 (2001) 1352-1361

[16] N. Megiddo and S. Modha, Outperforming LRU with an adaptive replacement cache algorithm, IEEE Computer (Apr. 2004) 4-11

[17] PeerApp Inc. <www.peerapp.com>

[18] S. Podlipnik and L. Böszörmenyi, A survey of web cache replacement strategies, ACM Computer Surveys (2003) 374-398

[19] T. Qiu et al., Modeling channel popularity dynamics in a large IPTV system, Proc. 11th ACM SIGMETRICS, Seattle, WA, USA (2009)

[20] Qwilt Inc. <www.qwilt.com>

[21] G. Rossini, D. Rossi, Coupling caching & forwarding: Benefits, analysis and implementation, Proc. ACM ICN, New York, USA (2014) 127–136

[22] R.K. Sitaraman et al., Overlay networks: An Akamai perspective, Chapter 16 in Advanced content delivery, streaming and cloud services, Eds. M. Pathan, et al., Wiley (2014) 307-328

[23] D. Stutzbach, S. Zhao, R. Rejaie, Characterizing files in modern Gnutella networks: A measurement study, Multimedia Systems 13/1 (2007) 35-50

[24] S. Traverso et al., Unraveling the impact of temp. and geo. locality in content caching systems, IEEE Trans. on Multimedia 17 (2015) 1839-54

[25] Wikipedia statistics and information available on
https://wikitech.wikimedia.org/wiki/Pageviews_API
https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia
https://meta.wikimedia.org/wiki/Wikimedia_servers
https://wikitech.wikimedia.org/wiki/Global_traffic_routing