Modeling and Performance Comparison of Caching Strategies for Popular Contents in Internet

Natalia M. Markovich[1(✉)], Vladimir Khrenov[1], and Udo R. Krieger[2]

[1] V.A. Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Profsoyuznaya Str. 65, 117997 Moscow, Russia
markovic@ipu.rssi.ru

[2] Fakultät WIAI, Otto-Friedrich-Universität, An der Weberei 5, 96047 Bamberg, Germany
udo.krieger@ieee.org

Abstract. The paper is devoted to caching of popular multimedia and Web contents in Internet. We study the Cluster Caching Rule (CCR) recently proposed by the authors. It is based on the idea to store only popular contents arising in clusters of related popularity processes. Such clusters defined as consecutive exceedances of popularity indices over a high threshold are caused by dependence in the inter-request times of the objects and, hence, their related popularity processes. We compare CCR with the well-known Time-To-Live (TTL) and Least-Recently-Used (LRU) caching schemes. We model the request process for objects as a mixture of Poisson and Markov processes with a heavy-tailed noise. We focus on the hit probability as a main characteristic of a caching rule and introduce cache effectiveness as a new metric. Then the dependence of the hit probability on the cache size is studied by simulation.

Keywords: Caching · Cluster Caching Rule · TTL · LRU · Hit/miss probability · Popularity process · Clusters of exceedances · Inter-request times

1 Introduction

Nowadays, caching of contents is intensively applied in the Internet to provide multimedia or Web objects on demand to the users with a minimal delay. The idea stems from computer systems where frequently demanded files have to be cached in a short memory to accelerate the exchange between the processor and the operative memory. In telecommunication systems this concept is used to keep the requested content in a cache, e.g. at an edge router in fog computing (cf. [19–21]), or a hierarchy of caches (cf. [3,4]). Numerous problems arising from the randomness of the inter-request time (IRT) sequences concern the optimal cache size, cache utilization and occupancy, and the replacement of objects within a cache to provide the fast availability of the requested content. The latter item is characterized by the hit/miss probability, i.e. the probability to find/miss a requested content in the cache.

V.M. Vishnevskiy et al. (Eds.): DCCN 2016, CCIS 678, pp. 47–56, 2016.
DOI: 10.1007/978-3-319-51917-3_5

Among these cache replacement rules the Least-Recently-Used (LRU) (cf. [1]), the Least-Frequently-Used (LFU) (cf. [2]) and the Time-to-Live (TTL) policy (cf. [3–5]) are the most popular schemes. Usually, the Independent Reference Model (IRM) that summarizes a number of assumptions is used to simplify the formulation of the hit/miss probability, the cache utilization and occupancy problems. According to IRM it is assumed that the inter-request times are independent and exponentially distributed (i.e. the request process is a Poisson renewal process), and that the popularity of contents or Web objects and content sizes are constant. The IRM implies a time and space locality regarding the object popularity. It should be noted that normally a non-Poisson renewal process model cannot capture the superposition of request processes that arise in cache networks (cf. [3]).

Not much work has been done when the IRM model is not appropriate. Then the IRT sequence may be correlated, heavy-tailed and non-stationary. Our first objective is to show how one can handle the caching problem in this case and what is the impact of such conditions on the effectiveness and utilization of a cache. Correlated IRTs are particularly realistic if some content has become very popular and many users are interested in it. Therefore, such correlations generate clusters of peaks of the popularity index. Following [6] we determine the cluster as a conglomerate of consecutive exceedances of the popularity process over a threshold between two consecutive non-exceedances. A cluster structure of the popularity process is shown in Fig. 1.

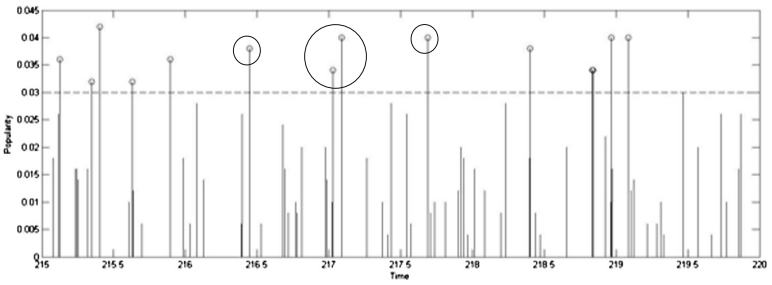

Fig. 1. The sequence of content popularity against the time including some indicated clusters of exceedances over a given threshold.

We focus on the Cluster Caching Rule (CCR) policy proposed in [7] and studied in [8]. Dealing with a single cache we propose here an *effectiveness of a cache* as new caching metric. It is defined as total popularity of objects placed in the cache at time t. The second objective is given by the analysis and comparison of the CCR, TTL and LRU rules by a simulation study. Both the CCR and TTL rule use *timers* as tuning knobs for individual objects to stay in the cache, but they apply different arguments. We propose to select the TTL timers depending on the popularity of the cached objects.

The paper is organized as follows. In Sect. 2 related work is discussed. In Sect. 3 we propose the effectiveness of a cache as characteristic metric. In Sect. 4 we modify the TTL rule regarding the specific TTL timers which depend on the popularity indices. Moreover, we compare the hit probabilities of the CCR, LRU and TTL rules depending on the cache size and the TTL timer selection by simulation. The results are summarized in the Conclusion.

2 Related Work

Cache replacement schemes can be split into capacity-driven and TTL-based policies (cf. [9]). The hit (or miss) probability determines the long-term frequency to find (or not to find) a requested object in the cache. The LRU and LFU policies belong to the capacity-driven group since objects are evicted from the cache by arrivals of those objects not yet stored. According to LRU a new requested object is placed into the cache and the least recently requested object is evicted from the cache. In case the requested object is found in the cache, it is put on the first position while the residual cache contents is shifted upwards. According to the TTL policy objects are evicted according to individual timers, i.e. life times to be in cache (cf. [3]). It was found that LFU is better than LRU (cf. [10]). Thus, modifications of LRU were proposed like persistent-access-caching (PAC) to improve its miss probability (cf. [11]).

The CCR policy [7] is related to a popularity oriented, threshold-driven policy. It allows to cache only those contents corresponding to related popularity clusters, i.e. those objects are cached whose popularity index exceeds a sufficiently high threshold u. The hit probability is then determined as the probability to enter the cluster and the time of an object to stay in the cache is determined by the duration of consecutive clusters containing that object and the corresponding inter-cluster times (see Fig. 2). The CCR scheme provides some kind of congestion control that allows to drive cache utilization. The threshold u determines the popularity level which is exceeded and impacts on the cluster sizes of the popularity process. CCR is in a way similar to LFU where only popular objects may be placed in the cache. Caching only frequently referenced objects has also been developed as central processing unit (CPU) approach in [1].

Regarding the stochastic analysis of caching rules for *correlated request processes with heavy tails* not much research has been done yet. Poisson arrival processes were considered in [12–14] with light- and heavy-tailed request rates λ_i, i.e. $\lambda_i \sim c\exp(-\xi i^\beta)$ for $i = 1, 2, \ldots$ with $c, \xi, \beta > 0$ and $\lambda_i \sim c/i^\alpha$ for $i = 1, 2, \ldots$ with $\alpha > 1$, $c > 0$, respectively. The miss probability of the LRU policy was shown to decrease following a power law or exponentially, respectively, for heavy- and light-tailed λ_i as the cache size C tends to infinity. It was derived that the correlation does not impact the miss probability for unlimited cache size. Markov arrival processes (MAPs) were also used to model correlated requests (cf. [3]), since they are self-contained regarding superposition. Regarding the LRU strategy and moderate cache sizes, non-stationary and dependent request processes and the average miss probability were considered as input and metric in [15].

Cache utilization determines an important metric and raises several issues. To optimize cache utilization based on TTL policies, it was proposed in [16] to maximize the sum of the utilities of all objects regarding the TTL timers. Therein, each content item is associated with a utility metric that is a function of the corresponding content hit probability. The latter approach assumes a Poisson renewal process as request model. In [7] the utilization with regard to the CCR strategy has been determined by the ratio of the cluster and the cache sizes where the cluster implies a set of consecutive exceedances of the popularity index over a sufficiently high threshold. Then the average cache utilization was considered both for fixed and random object sizes.

3 Effectiveness of the Cluster Caching Rule

The analysis of real traces has shown that about 70% of contents in caches is requested only once. It translates into an even higher miss ratio of 0.88 (cf. [1]). The LRU and TTL cache policies do not prevent to place unpopular contents in the cache. To prevent caching of a large portion of rarely requested objects, we propose to maximize the *effectiveness* of a cache. It is reflected by the new metric

$$e(t) = \sum_{i=1}^{C} p_i(t) \mathbb{1}\{i\text{th object } o_{j_i} \text{ from the catalog is in the cache at time t}\}.$$

We assume that all objects $\{o_j \mid j \in M\}, M = \{1, \ldots, N\}$ in the catalog have equal size s and $\widehat{C} = C \cdot s$ is the cache size. N denotes the size of the catalog. $p_i(t)$ is the popularity of the ith object o_{j_i} in the cache at epoch t. $e(t)$ indicates the total popularity of all those objects $\{o_{j_1}, \ldots o_{j_C}\}$ stored in the cache at time t. It holds $j_C \leq C$ since the cache may not be full. According to the CCR policy, the ith object o_{j_i} may be placed in the cache if its popularity $p_i(t)$ at time t exceeds a given threshold u.

As the cache load is provided by clusters of highly popular objects, their indices $p_i(t)$ may belong only to one cluster. This means that the number of cached objects is limited by the cluster size $T_2(u)$ or more exactly by the maximal cluster size. The notion of the cluster size of a stationary process $\{X_t\}_{t\geq 1}$

$$T_2(u) = \min\{j \geq 1 : L_{1,j} > u, X_{j+1} \leq u | X_1 \leq u\},$$

where $M_{1,j} = \max\{X_2, \ldots, X_j\}$, $M_{1,1} = -\infty$, $L_{1,j} = \min\{X_2, \ldots, X_j\}$, $L_{1,1} = +\infty$ is mentioned in [6,7] following [17]. Regarding the CCR policy, we then get the effectiveness

$$\begin{aligned} e_u(t) &= \sum_{i=1}^{C} p_i(t) \mathbb{P}\{p_i(t) > u | i\text{th object is in the latest cluster at time t}\} \\ &\leq \sum_{i=1}^{j} p_i(t) \mathbb{P}\{T_2(u) = j\} \end{aligned} \tag{1}$$

where $j \leq C$ is the observed cluster size. In case $j > C$ we can load the rest of those objects in the next cache of a cache hierarchy or increase u to decrease the cluster size.

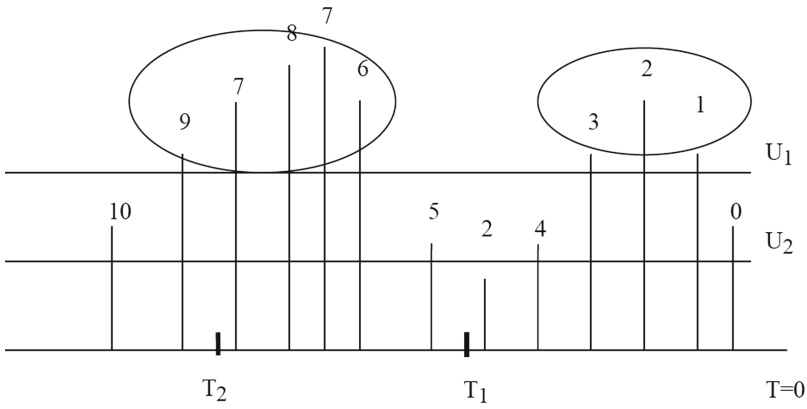

Fig. 2. Illustration of the CCR caching mechanism and the popularity clustering for different threshold values U_1 and U_2 over time.

Example 1. Figure 2 illustrates the dynamics of CCR caching and the calculation of the effectiveness. At time T_1 the cache contains objects with numbers 1, 2 and 3 because their popularity exceeds the threshold U_1. If U_2 were the threshold, the objects with numbers $0-4$ would be cached. Let us consider the threshold U_1. The next cluster begins at the object with number 6. The popularity of the object 2 decreases and it falls between two clusters. Nevertheless, at time T_1 it remains in the cache. In the second cluster the object with number 7 occurs twice. At time T_2 we have the objects with numbers $6-8$ in the cache. The objects $1-3$ are evicted from the cache. Hence, the effectiveness at time T_1 is calculated as the sum of the popularity of objects $1-3$ and at time T_2 by means of the objects $6-8$.

The probability $\mathbb{P}\{T_2(u) = j\}$ in (1) does not take into account possible repetitions of the same objects in the popularity clusters. Therefore, it provides an upper bound of the real effectiveness.

The effectiveness metric $e_u(t)$ in (1) is driven by u. We can find such u that provides a maximal value $e_u(t)$ for a fixed time t. To this end, let us assume that the objects' popularity is determined by Zipf's law, i.e. $p_i \sim \chi/i^{\alpha}$, where $\chi > 0$ is a constant. $\alpha > 0$ is the tail index. It shows the heaviness of the tail of the popularity distribution. As the popularity index may change over time, we can take $p_i(t) \sim \chi/i^{\alpha(t)}$.

Regarding a sequence of increasing thresholds $\{u_n\}_{n\geq 1}$, the probability of $T_2(u_n)$ derived in [6] satisfies for each $\varepsilon > 0$ and some n_ε and $j_0(n_\varepsilon)$ the following expression

$$| \mathbb{P}\{T_2(x_{\rho_n}) = j\}/(\theta^2 q_n (1-q_n)^{(j-1)\theta}) - 1 | < \varepsilon$$

for all $n > n_\varepsilon$ and sufficiently large j, i.e. $j > j_0(n_\varepsilon)$. Here high quantiles $\{x_{\rho_n}\}$ of the common popularity process of all objects in the catalog w.r.t. the levels $q_n = 1 - \rho_n$, $\rho_n \sim 1/n$ are taken as thresholds $\{u_n\}$. $\theta \in [0,1]$ is the dependence measure of the popularity process called extremal index [18]. The reciprocal $1/\theta$ approximates the mean cluster size of exceedances over the threshold $u = u_n$. By (1) and an approximation of the Rieman Zeta function for $\alpha(t) > 0, \alpha(t) \neq 1$, we get the total popularity of the j objects placed in the cache of size $\widehat{C} = C \cdot s$ in terms of

$$e_q(t) \approx \theta^2 q (1-q)^{(j-1)\theta} \sum_{i=1}^{j} \frac{\chi}{i^{\alpha(t)}} \approx \chi \theta^2 q (1-q)^{(j-1)\theta} \frac{j^{1-\alpha(t)} - 1}{1 - \alpha(t)}. \tag{2}$$

As the quantile level q represents now the threshold u, one can find $q = 1/(1 + (j-1)\theta)$ that maximizes $e_q(t)$. In Fig. 3 $e_q(t)$ is depicted for a fixed time t, i.e. $\alpha(t) = \alpha$. As Zipf's model may fit the popularity not accurately enough for samples of moderate size, we can estimate the popularity of the ith object o_{j_i} at stopping time t by [7]

$$p_i(t) = J_{i,t}/N_t. \tag{3}$$

Here $J_{i,t}$ and N_t denote the number of requests for the ith object o_{j_i} and for all objects $o_j, j \in M$ in the catalog at time t, respectively, that progress in time. The cluster size probability can be evaluated as ratio of the number of requests R_t with popularity exceedances over u to the total number of requests N_t at time t. Here, R_t contains only exceedances corresponding to different objects falling in the clusters. Then we get from (1) the empirical effectiveness

$$e_u(t) = \left[R_t/{N_t}^2\right] \sum_{i=1}^{C} J_{i,t}.$$

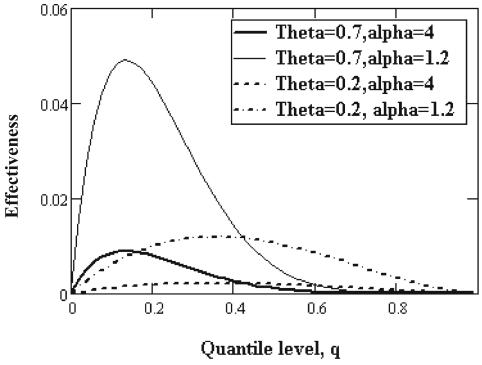

Fig. 3. Effectiveness (2) with $C = j = 10$ for the CCR policy against the quantile level q of the extremal index $\theta \in \{0.2, 0.7\}$ and the tail index $\alpha \in \{1.2, 4\}$, where $q \in \{0.137, 0.357\}$ corresponds to the maximal effectiveness.

Thereby, formula (2) provides the parametric model taking into account the heaviness of the tail in terms of $\alpha(t)$ and the dependence structure by θ.

An increasing level u induces clusters with smaller sizes. It may lead to the necessity to select a smaller cache size or to a less efficient utilization of the cache.

4 Performance Comparison of Different Caching Rules

We compare the CCR, LRU and TTL caching rules by simulation. Following [8] we use a mixture of the Moving Maxima (MM) and the Poisson renewal processes to model a common IRT process regarding all objects of the catalog of different types.

The MM process $\{\tau_{i,t}\}$ as IRT model of the ith object type satisfies

$$\tau_{i,t} = \max_{j=0,\dots,m_i} \{\alpha_j Z_{t-j}\}, \quad t \in \mathbb{Z},$$

with nonnegative constants $\{\alpha_j\}$ such that $\sum_{j=0}^{m_i} \alpha_j = 1$ and iid standard Fréchet distributed r.v.s $\{Z_t\}$ with distribution function $F(x) = \mathbb{P}\{Z_i \le u\} = e^{-1/u}$. The distribution of $\tau_{i,t}$ is also Fréchet. The MM process is a m_i-dependent Markov chain where m_i determines the popularity duration. The MM process models IRTs of short-term news that are of public interest for a limited time. The Poisson process with intensity λ_i models objects like scientific and culture articles which may attract interest within a long time independently. Each object of equal size $s = 1$ from the catalog has an own $(m_i, \{\alpha_j\})$ or λ_i value as unique IRT model parameter.

The MM processes generate the correlation and the cluster structure of such common IRT process that has been generated here by 90% MM and 10% Poisson renewal processes. The corresponding popularity process that is the popularity $p_i(t)$ of each requested object o_{j_i} calculated by (3) is given in Fig. 1. In (3) $J_{i,t}$ is calculated in a cross-window with $N_t = 300$ requests. The number of objects in the catalog was taken as $N = 100$.

We compare the CCR, the LRU and the TTL policies for such simulated IRT processes. For each object o_{j_i} we propose TTL timers $\{t_i\}$ depending on its popularity index $p_i(t)$ and the mean IRT $\mathbb{E}(Y_i)$ of the overall IRT process, i.e.

$$t_i = h\ \mathbb{E}(Y_i)\ p_i(t), \quad 0 < h < \infty. \tag{4}$$

h is a scalability parameter. The TTL timers are larger for highly popular objects. In (4) t_i is proportional to the popularity of the ith object in $[0, t]$.

In Fig. 4 we estimate the extremal index θ of the popularity process by the intervals estimator proposed in [17]. This allows us to estimate the effectiveness (2) and the cache size as the reciprocal $C = 1/\theta$ equal to the mean cluster size as proposed in [7]. Taking $\widehat{\theta} = 0.22$ it is easy to calculate the approximate mean cache size $\widehat{C} = 5$.

In Fig. 5 we show the hit probabilities for the TTL, LRU and CCR policies depending on the cache size C for $s = 1$. The hit probability is estimated as

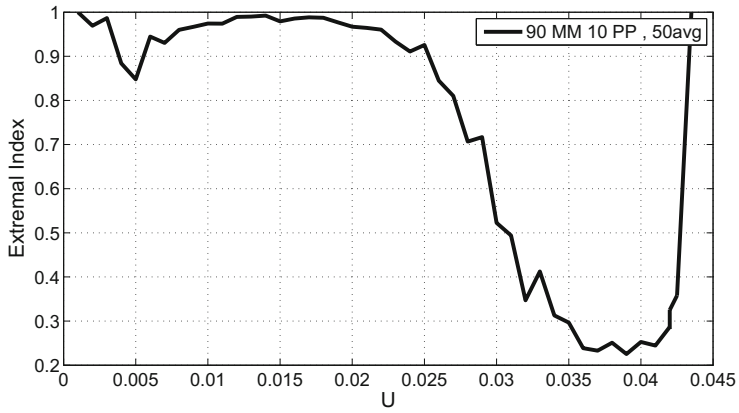

Fig. 4. The intervals estimate $\widehat{\theta}$ of the extremal index averaged over 50 samples against the threshold u: the estimate $\widehat{\theta} = 0.22$ corresponds to the stability interval of the plot by threshold u.

the ratio of the number of requests hitting the cache and the total number of requests. For small cache sizes the best hit probability is provided by the CCR scheme with a threshold u corresponding to the stability interval of the plot $(u, \widehat{\theta})$ and both the TTL and CCR work similar if h and u are relatively small. Small u generates large clusters. Then the CCR stores more objects in the same manner as TTL irrespectively of their popularity processes. If h and u are small, then the inter-cluster time for large clusters is of similar small scale as the TTL timers. For large caches and long timers TTL is better than CCR. This means a long-term placement of many objects in the large cache which is not effective. For large caches the CCR hit probability reaches a stability level

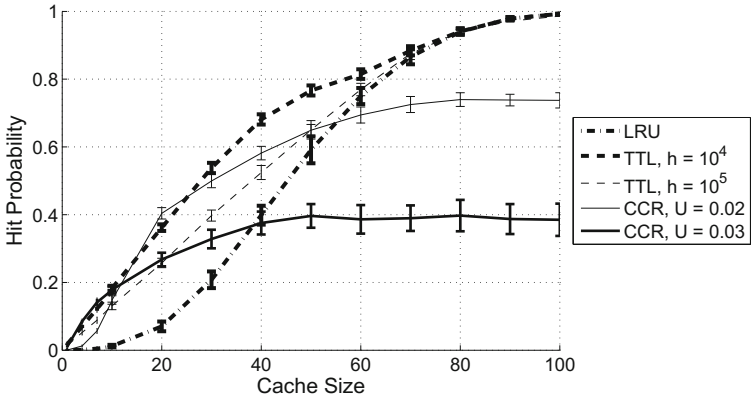

Fig. 5. The hit probabilities of the CCR, LRU and the TTL policies averaged over 50 samples against the cache size C, where horizontal lines indicate standard deviations.

that is lower than the corresponding TTL value due to the limited cluster size and the impossibility to store a larger number of objects than the cluster size. A minimal C corresponding to the stability level of the hit probability may be taken as a sufficient cache size.

5 Conclusion

The paper addresses the caching of popular multimedia and Web contents in Internet. We have extended the investigation of the Cluster Caching Rule (CCR) recently proposed in [7,8]. Assuming correlated inter-request time processes and fixed object sizes, we have studied here the caching of popular contents when the popularity of the stored objects may change over time. The LRU, CCR and TTL caching rules have been compared by a simulation study.

The following results have been obtained:

1. cache effectiveness has been introduced as new quality metrics;
2. regarding a TTL based policy, TTL timers based on popularity indices have been proposed;
3. the CCR policy has a better hit probability than TTL regarding relatively small cache sizes and thresholds u corresponding to the stability interval of the extremal index plot (u, θ);
4. the LRU policy is worse than both the CCR and TTL rule when the cache size is moderate and it works similar to TTL for large caches.

Regarding a fog computing environment based on interconnected powerful SBC boards (cf. [19,20]), optimized caching strategies for popular objects that implement the sketched approaches on a small memory are currently a very important research issue (cf. [21]). Consequently, the adoption of a dynamic version of the proposed CCR policy is a topic of our future research.

Acknowledgments. The first author acknowledges the financial support by DAAD scholarship 91619901.

References

1. Che, H., Tung, Y., Wang, Z.: Hierarchical web caching systems: modeling, design and experimental results. IEEE JSAC **20**(7), 1305–1314 (2002)
2. Lee, D., Choi, J., Kim, J.-H., Noh, S.H., Min, S.L., Cho, Y., Kim, C.S.: LRFU: a spectrum of policies that subsumes the least recently used and least frequently used policies. IEEE Trans. Comput. **50**(12), 1352–1362 (2001)
3. Berger, D.S., Gland, P., Singla, S., Ciucu, F.: Exact analysis of TTL cache networks: the case of caching policies driven by stopping times. In: 2014 ACM International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2014, pp. 595–596 (2014)
4. Fofack, N.C., Nain, P., Neglia, G., Towsley, D.: Analysis of TTL-based cache networks. In: 6th International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS), pp. 1–10 (2012)

5. Friecker, C., Robert, P., Roberts, J.: A versatile and accurate approximation for LRU cache performance. In: Proceedings of ITC 2012, pp. 1–8 (2012)
6. Markovich, N.M.: Modeling clusters of extreme values. Extremes **17**(1), 97–125 (2014)
7. Markovich, N.: A cluster caching rule in next generation networks. In: Vishnevsky, V., Kozyrev, D. (eds.) DCCN 2015. CCIS, vol. 601, pp. 305–313. Springer, Heidelberg (2016). doi:10.1007/978-3-319-30843-2_32
8. Markovich, N.M., Krieger, U.R.: A caching policy driven by clusters of high popularity. In: 7th IEEE International Workshop on TRaffic Analysis and Characterization (TRAC 2016), 5–9 September, Paphos, Cyprus (2016)
9. Rizzo, L., Vicisano, L.: Replacement policies for a proxy cache. IEEE/ACM Trans. Netw. **8**(2), 158–170 (2000)
10. Breslau, L., Cao, P., Fan, L., Phillips, G., Shenker, S.: Web caching and Zipf-like distributions: evidence and implications. In: IEEE Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 1999), vol. 1, pp. 126–134 (1999)
11. Jelenković, P.R., Radovanović, A.: The persistent-access-caching algorithms. Random Struct. Algorithms **33**, 219–251 (2008)
12. Jelenković, P.R.: Asymptotic approximation of the move-to-front search cost distribution and least-recently-used caching fault probabilities. Ann. Appl. Probab. **9**, 430–464 (1999)
13. Jelenković, P.R., Radovanović, A.: Least-recently-used caching with dependent requests. Theor. Comput. Sci. **326**(1–3), 293–327 (2004)
14. Jelenković, P.R., Radovanović, A.: Asymptotic optimality of the static frequency caching in the presence of correlated requests. Oper. Res. Lett. **37**(5), 307–311 (2009)
15. Osogami, T.: A fluid limit for a cache algorithm with general request processes. Adv. Appl. Probab. **42**, 816–833 (2010)
16. Dehghan, M., Massoulie, L., Towsley, D., Menasche, D., Tay, Y.C.: A utility optimization approach to network cache design, pp. 1–11 (2016). arXiv: 1601.06838v1
17. Ferro, C.A.T., Segers, J.: Inference for clusters of extreme values. J. Roy. Statist. Soc. Ser. B **65**, 545–556 (2003)
18. Leadbetter, M.R., Lingren, G., Rootzén, H.: Extremes and Related Properties of Random Sequence and Processes. Springer, Heidelberg (1983)
19. Großmann, M., Eiermann, A., Renner, M.: Hypriot cluster lab: an ARM-powered cloud solution utilizing docker. In: 23rd International Conference on Telecommunications (ICT 2016), 16–18 May, Thessaloniki, Greece (2016)
20. Großmann, M., Eiermann, A.: Security of distributed container based service clustering with hypriot cluster lab. In: Proceedings of ITC 28, September 12–16, Würzburg, Germany (2016)
21. Pahl, C., Lee, B.: Containers and clusters for edge cloud architectures - a technology review. In: 3rd International Conference on Future Internet of Things and Cloud (FiCloud), 24–26 August 2015, pp. 379–386 (2015)