Analysis of the LRU Cache StartUp Phase and Convergence Time and Error Bounds on Approximations by Fagin and Che

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

Konstantinos Ntougias
University of Cyprus
Cyprus, Greece
ntougias.konstantinos@ucy.ac.cy

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

Oliver Hohlfeld
Brandenburg Univ. of Tech.
Cottbus, Germany
oliver.hohlfeld@b-tu.de

Abstract – **We compare exact and approximate performance evaluation methods of the Least Recently Used (LRU) caching strategy, which is widely applied in local caches and in distributed web cache architectures in core and edge networks.**

Based on the independent reference model (IRM), the LRU startup behaviour and convergence time (CT) is derived. The result is related to hit ratio approximations by Fagin and Che et al. We evaluate the precision of the approximations by identifying maximum errors, which are shown to decrease with the cache size.

For different object sizes, we extend the analytical LRU hit ratio formula, which is tractable for small caches. The Che and CT approximations are subject to larger deviations for high variability of the object sizes and small caches due to partially unused cache space. We propose an estimation scheme for the fraction of unused LRU cache space, which is shown to improve the accuracy.

Keywords - LRU cache, hit ratio curve (HRC), LRU convergence time analysis, variable data size, Che and CT approximation, quantitative deviation study, independent request model (IRM)

I. Introduction: Analysis of Caching Methods

Web caching is essential for efficient content delivery, streaming and other services. Content delivery networks, cloud computing and information centric networking rely on distributed architectures, which shorten the transport paths and delays by data transfers from caches close to the requesting users [2][8][10][15][19][24]. The caching strategy selects the data to be stored, which is crucial for the caching performance. The cache hit ratio, i.e. the fraction of requests that can be served from a cache, is the main performance measure, from which quality of service gains can be derived in terms of delay and load reduction. Demands for ultra-low delay in 5G/6G networks strengthen the relevance of edge cache servers [23].

The focus of this work is on the analysis of the Least Recently Used (LRU) caching strategy [21], which is widely applied in local caches for CPU and database processing. In web caching, content selection schemes with awareness of object properties, such as the size, popularity and specific caching value often outperform LRU [1][10][14][22][25]. Nonetheless, LRU is the standard reference also for the analysis of distributed web caching networks [2][8][19][24].

An exact analysis of the steady state LRU hit ratio under the independent reference model (IRM) was derived by [18], whose computation effort is exponentially increasing with the cache size. Therefore, approximation approaches are frequently used as a simpler and scalable LRU hit ratio estimate [8][9][11][12].

Our initial focus is on the LRU hit ratio during cache filling phases under IRM request pattern starting from an empty cache. This includes the convergence time (CT) of LRU to steady state, because LRU enters steady state behavior as soon as the cache is filled. The study is also relevant for many other caching strategies, which do not evict objects until the cache is filled and thus follow the same cache filling process as LRU. We show that FIFO and RANDOM strategies achieve the LRU IRM hit ratio level when the cache is filled, from which they are afterwards declining to a common FIFO and RANDOM hit ratio [13] in a second transient phase.

Moreover, the mean convergence time $\overline{CT}_{IRM}^{LRU}$ is shown to be closely related to the "*characteristic time*", as considered in Che's approximation of the LRU hit ratio [8][12] and leads to an equivalent approach as proposed by Fagin [11] via the "*expected working-set miss ratio with window size T*".

A set of recent studies [4][5][12][17][26] confirm an asymptotic convergence of both approximations for large caches and object catalogues. However, such best-case behavior results do not lead to conclusions on the real deviations, which are estimated by Che et al. based on simulation studies: "*This solution is tested against simulation results, which shows that the solution is highly accurate with a maximum error less than 2%.*"

We perform an exhaustive quantitative evaluation for cases up to a limited cache size, which identifies a special format of popularity profiles that leads to maximum deviations of Fagin's and Che's approach. In this way, we find extreme cases of $> 8\%$ error, but the results strongly suggest that the maximum absolute error is decreasing with the cache size M.

In a 3rd part, we include objects of different size, for which we provide an extension of the exact LRU hit ratio formula [18]. Moreover, we study direct extensions of the approximations by Fagin [11] and Che et al. [8] for variable object sizes. We show that the basic approach proposed by [12] in Section 3.2, is often subject to large deviations for small caches and/or high variance of the object sizes [1][3]. We identify oversize objects as well as unused cache space (*UCS*) as components that add to deviations, which can lead to zigzag shaped hit ratio curves. We derive an estimation scheme for the mean *UCS*, which essentially improves the LRU hit ratio approximation. As the main new contributions of this work, we provide

- an analysis of the LRU convergence time and the performance of LRU and other caching strategies during cache filling phases,
- a quantitative analysis of the deviations of LRU hit ratio approximations by Fagin [11] and Che et al. [8], which identifies their worst deviation cases for small cache size M and confirms bounds on their accuracy depending on M,
- extensions of the exact LRU hit ratio solution and approximations for caches with objects of different size.

We start with exact analysis results of the LRU hit ratio performance in Section II.A - II.B., followed by an evaluation of

ISBN 978-3-903176-49-2

LRU cache filling phases and the convergence time distribution to steady state in Sections II.C - II.E. Approximations approaches of the LRU hit ratio are compared in Sections II.F - 0. Extensions of those approximations for objects of different size are studied in Section III, including an estimation of the unused cache space for improving the precision.

II. LRU Steady State & Convergence Time Analysis

As a basic performance result for LRU caches, W.F. King [18] derived a steady state hit ratio formula assuming independent requests. The independent request model (IRM) is characterized by a catalogue of N objects $O_1, \ldots, O_N$, which are referenced with probabilities $p_1, \ldots, p_N$ for each request independent of previous references. Moreover, a fixed cache size for M objects of unit size is assumed. IRM request pattern with Zipf distributed object popularity has been confirmed manifold as a realistic model for web request traces [7][15][24]. Moderate correlation among requests and changes in the working set of objects are also relevant [15][24][27].

A. IRM Hit Ratio Formula for LRU Caches

Steady state probabilities for the content in an LRU cache can be derived in a top-down approach along the stack positions following the least recently used (LRU) cache eviction rule: The top position of the LRU stack is occupied by the most recently requested object, such that we find object O_j with probability p_j on top. The next request beforehand, which referred to another object, fills the 2nd LRU stack position, such that an object $O_k \neq O_j$ is found there with probability $p_k/(1-p_j)$. Another object $O_l \neq O_j, O_k$ is found in the 3rd position with probability $p_l/(1-p_j-p_k)$. Then the IRM steady state probabilities $p_{LRU}(O_{k_1}, \ldots, O_{k_M})$ for the content in an LRU cache of size M and the hit ratio h_{IRM}^{LRU} are given by [2][9][18][20] ($\forall j \neq l: k_j \neq k_l; \forall j: k_j \neq n$):

$$p_{LRU}(O_{k_1}, \cdots, O_{k_M}) = p_{k_1} \frac{p_{k_2}}{1-p_{k_1}} \cdots \frac{p_{k_M}}{1-p_{k_1}-\cdots-p_{k_{M-1}}}; \quad (1)$$

$$h_{IRM}^{LRU} = \sum_{k_1,\ldots,k_M=1}^{N} p_{LRU}(O_{k_1}, \ldots, O_{k_M}) \sum_{j=1}^{M} p_{k_j} \quad (2)$$

$$= \sum_{n=1}^{N} p_n \left(p_n + \sum_{m=1}^{M-1} \sum_{k_1,\ldots,k_m=1}^{N} p_{LRU}(O_{k_1}, \ldots, O_{k_m}, O_n)\right).$$

The latter representation of h_{IRM}^{LRU} in the 2nd line of (2) distinguishes cases to find O_n in the positions $m+1 = 2, \ldots, M$ of the stack. The first term $p_n \cdot p_n$ represents the contribution of hits on O_n in the top stack position. Note that the probabilities $p_{LRU}(O_{k_1}, \ldots, O_{k_m})$ are valid not only for a set of M objects that fills the cache, but also for stack rankings of the top m objects on the entire range $m = 1, \ldots, N$ for any IRM request sequence with references to m different objects. However, $N!/(N-M)!$ summands are involved in the hit ratio formula, which is tractable only for small cache sizes.

B. LRU Hit Ratio for Objects of Different Size

Next, we consider an LRU cache for storing objects O_k of different sizes s_k, where the fixed cache size M is measured in Byte. We exclude objects, which do not fit into the cache, i.e. we assume $s_k \leq M$ for $k = 1, \ldots, N$ and we still assume IRM requests with probabilities p_k. Upon a cache miss, objects are evicted due to the LRU principle, until the requested object fits into the cache. In this way, a requested external object may be cached without an eviction or after several evictions, depending on its size and the free space in the cache.

The steady state probabilities (1-2) for the sequence of the top m objects of the LRU stack are still valid for variable object sizes on the entire range $m = 1, \ldots, N$. Therefore, the hit ratio formula can be straightforwardly extended for objects of different size. We refer to the representation in the 2nd line of (2) with a summand for each stack position, which allows to restrict to sets of objects that fit into the cache. Then we obtain the following extended LRU cache hit ratio formula including objects of different sizes $s_1, \ldots, s_N$ ($\forall j \neq l: k_j \neq k_l; \forall j: k_j \neq n$):

$$h_{IRM}^{LRU} = \sum_{n=1}^{N} p_n \left(p_n + \sum_{m=1}^{N-1} \sum_{\substack{k_1,\ldots,k_m=1 \\ s_{k_1}+\cdots+s_{k_m}+s_n \leq M}}^{N} p_{LRU}(O_{k_1}, \ldots, O_{k_m}, O_n)\right)$$

$$= \sum_{n=1}^{N} p_n^2 \left(1 + \sum_{m=1}^{N-1} \sum_{\substack{k_1,\ldots,k_m=1 \\ s_{k_1}+\cdots+s_{k_m}+s_n \leq M}}^{N} \prod_{j=1}^{m} \frac{p_{k_j}}{1-\sum_{i=1}^{j} p_{k_i}}\right). \quad (3)$$

C. Convergence and Mixing Time Estimates

Beyond the previous steady state results, the convergence or mixing time to steady state is an important performance criterion. Fast convergence means short time to adapt the cache content to changing working sets or changing object popularity in transient phases and for non-stationary request pattern.

In a recent study, Li et al. [20] derive formulas for the order of magnitude of the IRM mixing time of basic caching strategies in general, and especially for Zipf distributed requests. LRU mixing times are shown to outperform FIFO and RANDOM. The results [20] are extended to multi-segment caches.

D. LRU Cache Filling and Steady State Convergence Time

We confirm fast convergence speed of LRU under IRM request pattern not only in terms of the order of magnitude, but more directly by considering the development of the cache content during filling phases starting from an empty cache. We can make use of the fact that LRU is already in steady state behavior as soon as the cache is filled.

The probability $p_{Cache\text{-}Fill}(O_{k_1}, \ldots, O_{k_m})$ that $O_{k_1}, \ldots, O_{k_m}$ are the first m objects to enter an empty cache is again determined by the steady state result (1) for the content distribution in an LRU stack. We obtain $p_{Cache\text{-}Fill}(O_{k_1}) = p_{k_1}$ for O_{k_1} being the first object to enter an empty cache. When $O_{k_1}, \ldots, O_{k_{m-1}}$ are in the cache, O_{k_m} will enter as the next object with probability $p_{k_m}/(1-p_{k_1}-\cdots-p_{k_{m-1}})$. We conclude ($\forall j \neq l: k_j \neq k_l$):

$$\begin{aligned} &p_{Cache\text{-}Fill}(O_{k_1}, \cdots, O_{k_m}) \\ &= p_{Cache\text{-}Fill}(O_{k_1}, \cdots, O_{k_{m-1}}) \cdot p_{k_m}/(1-p_{k_1}-\cdots-p_{k_{m-1}}) \quad (4) \\ &= p_{k_1} \cdot (p_{k_2}/(1-p_{k_1})) \cdot \cdots \cdot p_{k_m}/(1-p_{k_1}-\cdots-p_{k_{m-1}}) \quad \Rightarrow \\ &p_{Cache\text{-}Fill}(O_{k_1}, \cdots, O_{k_m}) = p_{LRU}(O_{k_1}, \cdots, O_{k_m}) \quad \text{for} \quad m = 1, \ldots, M. \end{aligned}$$

In this way, the LRU cache content distribution (1) also characterizes cache filling phases: When m objects have entered the cache then the hit ratio of the next request is given by the LRU steady state hit ratio for a cache of size m. The hit ratio is

increasing with the filling level m towards the value for the maximum cache size M. The cache filling behavior is the same for other caching strategies such as FIFO, LFU, GreedyDual, Score-based methods etc. [10][14][22][25], because they differ only in the treatment of evictions, but no evictions are performed during filling phases when there is enough room for new objects.

In order to determine the LRU convergence time distribution, we define the probabilities $p_r(O_{k_1}, \ldots, O_{k_m})$ that the objects $O_{k_1}, \ldots, O_{k_m}$ have entered an initially empty cache during the first r requests. We start from $p_1(O_{k_j}) = p_{k_j}$ for $j = 1, \ldots, N$. The next request leaves the set of cached objects unchanged in case of a cache hit or, after a miss, a new object enters. We obtain an iterative scheme to compute $p_r(O_{k_1}, \ldots, O_{k_m})$ and finally the distribution $\text{Prob}\{CT_{IRM}^{LRU} = j\}$ of the LRU convergence time, corresponding to a partial coupon collection process ($\forall j \neq l$: $k_j \neq k_l$; $m \leq M$):

$$p_{j+1}(O_{k_1}, \ldots, O_{k_m}) =$$

$$p_j(O_{k_1}, \ldots, O_{k_m}) \sum_{\ell=1}^{m} p_{k_\ell} + \sum_{\substack{k_m=1 \\ \forall j<m:\, k_m \neq k_j}}^{N} p_j(O_{k_1}, \ldots, O_{k_{m-1}})\, p_{k_m}; \qquad (5)$$

$$\text{Prob}\{CT_{IRM}^{LRU} = j\} = \sum_{\substack{k_1, k_2, \ldots, k_{M-1}=1 \\ \forall j \neq l: k_j \neq k_l}}^{N} \sum_{\substack{k_M=1 \\ \forall j<M:\, k_M \neq k_j}}^{N} p_{j-1}(O_{k_1}, \ldots, O_{k_{M-1}})\, p_{k_M}. \qquad (6)$$

The equations (4 - 6) provide a basic result on LRU caching which seems to be new according to Wong et al. [28]: "The only work which the authors are aware of on transient cache startup was done by Bhide et. al. [6].". While Bhide et al. [6] are extending an approximation by Dan and Towsley [9] for the warmup time evaluation, Wong et al. [28] are deriving exact recursive equations for the number of distinct objects in the cache and the cache miss probability. The results in [28] are similar to (4 - 6), but the authors of [28] do not make use of the basic relationship (4), which extends the classical steady state LRU result [18] to the cache filling phases.

Although the analysis of the result (4 - 6) is presuming and starting from an empty cache, the LRU convergence time is independent of the initial cache content. The reason behind is the fact that only the most recent requests to M different objects determine the current LRU stack. The LRU caching process has no memory of what happened before the time span for the last accesses to M different objects, where all other objects are evicted, if they are not requested within this time span.

E. Comparison of LRU, FIFO and LFU convergence times

Next, we compare the cache filling phases of LRU and other strategies based on simulation results. Besides the LRU convergence time, our focus is on the development of the cache hit ratio in the filling phase and afterwards towards the steady state behaviour. We evaluate an example of Zipf distributed IRM requests [7] with $\beta = 1$

$$p_k = \alpha k^{-\beta} \quad \text{for } k = 1, \ldots, N; \quad \alpha = 1 / \sum_{k=1}^{N} k^{-\beta} \qquad (7)$$

for cache size $M = 1000$ and $N = 10^6$ objects. Figure 1 shows the development of the hit ratio of the r^{th} request during the filling phase starting from an empty cache. Each result represents the mean value of 1000 simulations.

In the example, simulated cache filling phases are lasting for a mean number of $\overline{CT}_{IRM}^{LRU} \approx 1501$ requests. In this phase, other considered strategies show the same behavior as LRU. When the cache is filled, LRU is in steady state and then keeps a constant hit ratio level, whereas other methods pass through a second transient phase from LRU to their own steady state behavior. While RANDOM and FIFO hit ratios decline from LRU level to a lower level, the LFU hit ratio is increasing towards the maximum IRM hit ratio in a long-lasting convergence process, which is still about 1% below the maximum level after 50 000 requests. The partition of the convergence time of FIFO, RANDOM and LFU policies in a first phase representing the LRU convergence time and an additional second transient phase confirms mixing times results by Li et al. [20], which also show that the LRU convergence is fastest.

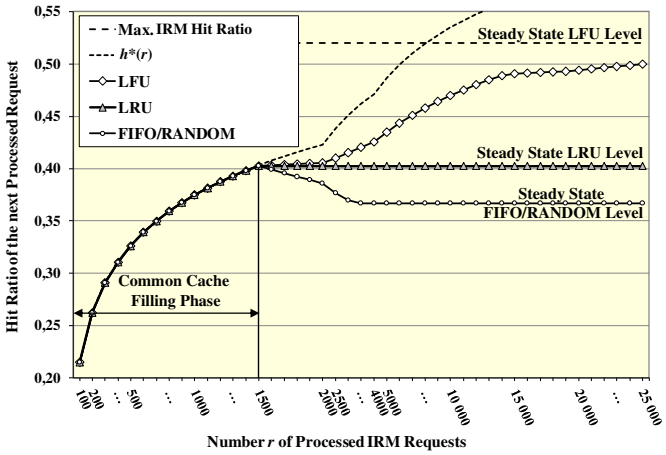

Figure 1: Hit ratio development in cache filling phases ($M = 1000$; $N = 10^6$; Zipf distributed requests with $\beta = 1$)

The maximum IRM hit ratio is shown by a horizontal dotted line. $h_{IRM}^{MAX} = p_1 + p_2 + \cdots + p_M$ is achieved, when M most popular objects $O_1, \ldots, O_M$ are cached, assuming that $p_1 \geq \ldots \geq p_N$. Moreover, Figure 1 includes a curve for a hit ratio bound $h^*(r)$ in the r^{th} request. We compute $h^*(r) = \Sigma_k\, p_k \cdot (1 - (1 - p_k)^{r-1})$, where the term $1 - (1 - p_k)^r$ represents the probability that an object O_k is requested and enters an empty LRU cache within the first r requests. The bound $h^*(r)$ is exact, if no evictions are encountered, i.e. for $r \leq M$ or for caches of unrestricted size. As compared to the simulated LRU hit ratio curve, $h^*(r)$ is confirmed to have negligible deviations for $r < \overline{CT}_{IRM}^{LRU} \approx 1501$. For larger r, $h^*(r)$ increasingly overestimates the LRU hit ratio because of evictions that are ignored in the $h^*(r)$ computation.

F. CT and Che Approximations of the LRU Cache Hit Ratio

The LRU steady state and convergence time analysis results of the previous Sections II.A - II.D involve sets of up to M cached objects and are tractable only for small caches. Therefore, approximations are relevant for tractable analysis, which have been proposed by Fagin [11] and Che et al. [8]. Both approaches are similar and they are again based on the computation of the mean LRU convergence time $\overline{CT}_{IRM}^{LRU}$.

Therefore, we observe that an object O_k is requested with probability $p_r(O_k) = 1 - (1 - p_k)^r$ within r requests. Then a mean number $\#_{Objects}(r) = \sum_{k=1}^{N} 1 - (1 - p_k)^r$ of objects has en-

tered an empty cache after r requests. LRU is converging, when $\#_{Objects}(r)$ approaches the cache size M. We conclude

$$M \approx \#_{Objects}(\overline{CT}_{IRM}^{LRU}) = \sum_{k=1}^{N} 1-(1-p_k)^{\overline{CT}_{IRM}^{LRU}}. \qquad (8)$$

The right side of (8) is monotonously increasing with $\overline{CT}_{IRM}^{LRU}$ from 0 to N, such that there is a unique solution for $\overline{CT}_{IRM}^{LRU}$. The same format (8) was proposed by Fagin [11] as the basis of an LRU hit ratio approximation, although not in the context of cache filling and LRU convergence time processes.

Finally, $h_{CT} = h^*(\overline{CT}_{IRM}^{LRU})$ is useful to estimate the LRU hit ratio when $\overline{CT}_{IRM}^{LRU}$ is determined via (8). We summarize the convergence time (CT) approximation of the LRU hit ratio:

$$M \approx \sum_{k=1}^{N} 1-(1-p_k)^{\overline{CT}_{IRM}^{LRU}};\ h_{CT}(O_k) = 1-(1-p_k)^{\overline{CT}_{IRM}^{LRU}};$$
$$h_{CT} = \sum_{k=1}^{N} p_k h_{CT}(O_k) = \sum_{k=1}^{N} p_k (1-(1-p_k)^{\overline{CT}_{IRM}^{LRU}}). \qquad (9)$$

The approach is equivalent to the approximation proposed by Fagin [11] as the "*expected working-set miss ratio*" without reference to the LRU convergence time. In the sequel, we show that this approach (9) is also closely related to Che's approximation [8] and we compare the accuracy of both approaches in a detailed quantitative study.

Che et al. [8] assume that requests for documents O_k follow a Poisson request model (PRM) with request rate λ_k. The PRM probability that O_k is requested during time interval ΔT is given by $1 - e^{-\lambda_k \Delta T}$. Poisson requests are memoryless and thus imply IRM requests, such that the next request refers to O_k with probability $p_k = \lambda_k/\lambda$, where $\lambda = \sum_k \lambda_k$. Then the mean number $\#_{Objects}(\Delta T)$ of different objects being requested during time ΔT is given by a sum of the probabilities $1 - e^{-\lambda_k \Delta T}$, that O_k is requested. When the mean sojourn time ΔT_{LRU} of an object in an LRU cache of size M is considered, $\#_{Objects}(\Delta T_{LRU})$ should be equal to M, which leads to an implicit relationship to obtain ΔT_{LRU} [8][12], similar to the derivation of (8 - 9):

$$\#_{Objects}(\Delta T_{LRU}) = \sum_{k=1}^{N} 1 - e^{-\lambda_k \Delta T_{LRU}} \approx M. \qquad (10)$$

After the mean sojourn time ΔT_{LRU} of objects in an LRU cache has been determined via (10), the hit ratio $h_{LRU}(O_k)$ per object is obtained as the probability that O_k is referenced again within ΔT_{LRU}, when O_k still resides in the cache:

$$h_{LRU}(O_k) \approx \text{Prob}\{\textit{Inter Request Time}(O_k) \le \Delta T_{LRU}\} = 1 - e^{-\lambda_k \Delta T_{LRU}}.$$

Finally, Che's approximation h_{Che} of the LRU hit ratio h_{LRU} is computed in two steps [8]: First, the solution $\Delta_{LRU} = \lambda \Delta T_{LRU}$ is determined from

$$M \approx \sum_{k=1}^{N} 1 - e^{-p_k \Delta_{LRU}}, \quad \text{where } p_k = \lambda_k/\lambda. \qquad (11)$$

Then the hit ratio is obtained per request: $h_{Che}(O_k) = 1 - e^{-p_k \Delta_{LRU}}$ and in total

$$h_{Che} = \sum_{k=1}^{N} p_k \cdot h_{CHE}(O_k) = \sum_{k=1}^{N} p_k (1 - e^{-p_k \Delta_{LRU}}). \qquad (12)$$

The approach (11 - 12) can be derived within the framework of Time-To-Live (TTL) caching, see e.g. Jiang et al. [17], where its scope is also widened beyond IRM request streams.

On the whole, the close relationship between the results (4-6) for the LRU convergence time CT_{IRM}^{LRU} and the approximation variants (8 - 9) and (11 - 12) of its mean $\overline{CT}_{IRM}^{LRU}$ shows that the notations of

- "*the expected working-set size*" by Fagin [11] and
- "*the characteristic time*" by Che et al. [8]

are equivalent, such that all those hit ratio approximations [8] [11] may simply and uniquely be referred to as approximations based on the mean LRU convergence time.

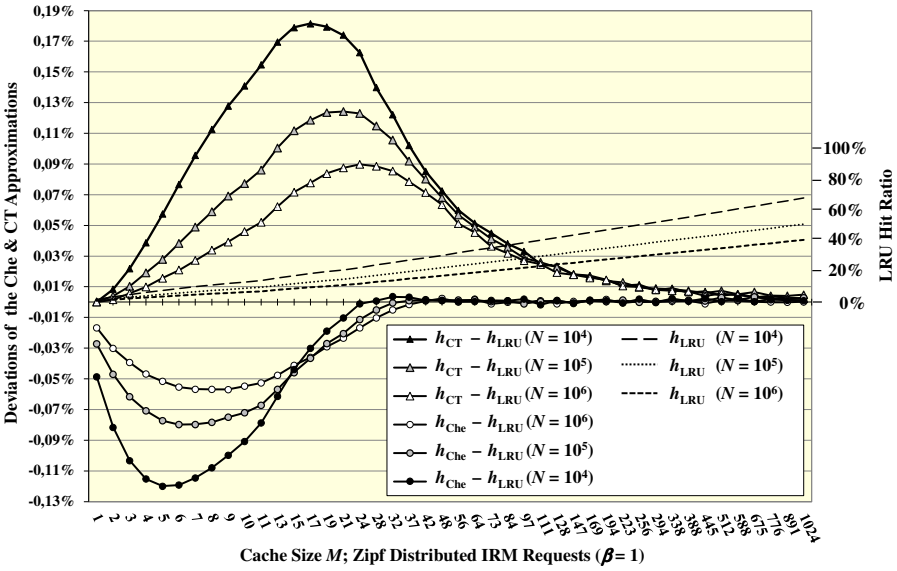

Figure 2: Deviation curves Δh_{Che} and Δh_{CT}

G. Quantitative Study of Deviations of the Approximations

Che's approximations (11-12) and the approach (9) are similar. However, they differ in the factor $1 - p_k$ in (9) being substituted with e^{-p_k} in (11-12). The derivation of $\overline{CT}_{IRM}^{LRU}$ follows a discrete IRM model at request instances, whereas Che's approach [8] [12] is transferring a continuous Poisson request model to IRM.

Next, we compare the deviations $\Delta h_{Che} = h_{Che} - h_{LRU}$ and $\Delta h_{CT} = h_{CT} - h_{LRU}$ for both approximations. In Figure 2, the LRU results are obtained by simulation of Zipf distributed IRM requests as defined in (7) with $\beta = 1$.

Three typical deviation curves of Δh_{Che} and Δh_{CT} for Zipf distributed IRM requests are shown for object catalogue sizes $N = 10^4$, 10^5 and 10^6.

All absolute deviations $|\Delta h_{Che}|$ and $|\Delta h_{CT}|$ are below 0.2% in these cases. Our LRU simulations are performed over 10^9 requests in each case, which leaves uncertainties in terms of 95% confidence intervals with a width of about $2 \cdot 10^{-5}$ [14].

Small deviations for Zipf distributed IRM requests similar to the curves of Figure 2 are also obtained by [14][24]. Moreover, recent analysis studies [4][5][12][26] show asymptotic exactness of Che's approximation for large M, N due to a statistical multiplexing effect. Then the number of different objects being requested in an interval ΔT_{LRU} is close to a Gaussian distribution [12], or an underlying coupon collection process is analysed in [26]. However, those results are not accompanied with bounds on the remaining deviations $|\Delta h_{Che}|$ and $|\Delta h_{CT}|$.

Therefore, we extend the deviation checks for different distribution types including geometric and linear distributions, in order to find the maximum deviations of h_{Che} and h_{CT}.

Finally, we checked all 204266 popularity distributions with $p_k = i_k/50$ for $k \le N \le 50$, where i_k is an integer, i.e., all distributions, whose request probabilities are multiples of 1/50. The LRU hit ratios for all cache sizes $M < N$ are obtained via simulation of 10^6 requests. The results indicate that maximum deviations $|\Delta h_{Che}|$ and $|\Delta h_{CT}|$ are encountered for distributions of n equally popular objects among many rarely referenced ones, i.e. for distributions of the type

$$p_1 = p_2 = \cdots = p_n = p/n; \quad p_{n+1} = \cdots = p_N = (1-p)/(N-n) \to 0; \quad (13)$$

for $N \to \infty$. In such cases, the central limit based asymptotes [4][12] do not apply, especially when n or M is small.

Figure 3 shows curves with the largest deviations that we obtained for M = 1, 2, 3 as peak values depending on p. The maximum deviations are decreasing with larger cache size M, except for $|\Delta h_{CT}| = 0$ for $M = 1$. We obtain the overall maxima

$$\max(|\Delta h_{Che}|) \approx 8.25\% \text{ for } M = n = 1;\ p \approx 0.845, \text{ and}$$

$$\max(|\Delta h_{CT}|) \approx 5.20\% \text{ for } M = 2,\ n = 1;\ p \approx 0.68.$$

The maximum deviations in Figure 3 can be determined analytically. The steady state LRU cache content for requests of the distribution type (13) is obtained similar to (1-2), yielding

$$h_{LRU}(n = 1, p, M) = p - p(1-p)^M; \quad (14)$$

$$h_{LRU}(n = 2, p, M) = p - p[(1-p)/(1-p/2)]^{M-1} + p^2(1-p)^{M-1}/2; \ \ldots$$

As general trends for the deviations of the LRU approximations we observe:

- Maximum deviations $|\Delta h_{Che}|$ and $|\Delta h_{CT}|$ are encountered for small cache sizes M.
- The h_{CT} result is exact for $M = 1$: $h_{CT} = h_{LRU} = \sum_{k=1}^{N} p_k^2$. This is a special case of (9) $M = 1 \Leftrightarrow \overline{CT}_{IRM}^{LRU} = 1$.
- In all simulation results we obtain
  $h_{Che} \le h_{CT}$ ($\Rightarrow$ for $M = 1$: $h_{Che} \le h_{LRU}$).
- In general, we experience $|\Delta h_{Che}|$ and $|\Delta h_{CT}|$ to decrease with smaller variance of the request distribution.

Both approaches h_{Che} and h_{CT} are exact for uniform popularity. Zipf distributed popularity is approaching a uniform shape for $\beta \to 0$ and the variance is decreasing for larger N [4][26]. Thus, the deviations are reducing for Zipf distributions when $\beta \to 0$ or $N \to \infty$.

With a closer look at quantitative results per cache size in the range $M = 1, \cdots, 10$, we obtain the maximum absolute deviations of Table 1. The CT approach overestimates the LRU hit ratio in cases of maximum deviations Δh_{CT}, which are decreasing with M and stay below 2% for $M > 5$. The number n of popular objects in extreme distributions is about half of M.

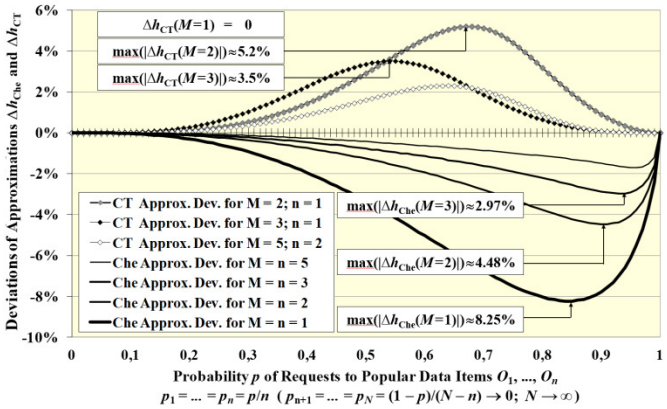

Figure 3: Maximum deviations of $|\Delta h_{Che}|$ and $|\Delta h_{CT}|$

Deviations of the Che approach are more balanced between positive and negative cases. For $M < 8$, maximum deviations Δh_{Che} are negative with $n = M$ in extreme cases. For $M \ge 8$, the maximum deviations are positive, but in general, the maximum deviations are decreasing with M and below 1% for $M > 8$.

Table 1: Maximum deviations of Δh_{Che} and Δh_{CT} for $M \le 10$

Maximum Deviations of the Che and CT approximation for Cache Sizes $M \le 10$ for Worst Cases Request Distributions of the Type of Equation (13)				
M	Parameters (13) n ‖ $p_1 = \cdots = p_n$	h_{IRM}^{LRU} ‖ Max. Che approx. deviation	Parameters (13) n ‖ $p_1 = \cdots = p_n$	h_{IRM}^{LRU} ‖ Max. CT approx. deviation
1	1 ‖ 0.845	0.706 ‖ –0.0825	CT approximation is exact	
2	2 ‖ 0.455	0.797 ‖ –0.0448	1 ‖ 0.630	0.597 ‖ +0.0521
3	3 ‖ 0.310	0.825 ‖ –0.0297	1 ‖ 0.540	0.488 ‖ +0.0353
4	4 ‖ 0.235	0.839 ‖ –0.0218	2 ‖ 0.360	0.665 ‖ +0.0282
5	5 ‖ 0.192	0.882 ‖ –0.0171	2 ‖ 0.315	0.580 ‖ +0.0231
6	6 ‖ 0.160	0.883 ‖ –0.0139	3 ‖ 0.247	0.695 ‖ +0.0199
7	7 ‖ 0.139	0.905 ‖ –0.0117	3 ‖ 0.227	0.638 ‖ +0.0172
8	6 ‖ 0.155	0.910 ‖ +0.0103	4 ‖ 0.187	0.710 ‖ +0.0154
9	7 ‖ 0.134	0.921 ‖ +0.0970	5 ‖ 0.158	0.750 ‖ +0.0139
10	8 ‖ 0.119	0.934 ‖ +0.0905	5 ‖ 0.150	0.711 ‖ +0.0126

All cases of maximum deviations in Table 1 are supported by the evaluation of 204266 popularity distributions with $p_k = i_k/50$. Each sorted list of distributions with largest absolute deviations for a specific cache size M starts with dozens or even hundreds of cases, which are closest to the type of equation (13) with the specific parameter set for each case as indicated in Table 1. Then there are n objects with request probabilities $p_k = i_k/50$ ($i_k \in \{1, \cdots, 50\}$) around the common value for $p_1 = \cdots = p_n$ given in Table 1 and $p_k = 0$ for $k > n$, or eventually $p_k = 1/50$ for one or a few more objects.

In summary, the quantitative study strongly suggests that the maximum deviations $|\Delta h_{Che}|$ and $|\Delta h_{CT}|$

- are encountered for the distribution type defined by (13),
- are decreasing with M,
- are bounded by $|\Delta h_{Che}| < 0.01$ and $|\Delta h_{CT}| < 0.13$ for $M \ge 10$.

However, our quantitative results do not provide coverage for a general confirmation of those extreme cases for large caches and leave a proof of the latter properties for future study.

III. LRU Approximations For Objects Of Different Size

The Che and CT approximations can be straightforwardly extended to objects of different size [12]. The LRU caching scheme with regard to individual object sizes s_k is described in Section II.B. The implicit relationships (8) and (11) include variable object sizes s_k in the following format [12]:

$$M=\sum_{k=1}^{N} s_k(1-e^{-p_k \Delta_{\mathrm{LRU}}});\quad M=\sum_{k=1}^{N} s_k(1-(1-p_k)^{\overline{CT}_{IRM}^{LRU}}). \quad (15)$$

Together with (15), the hit ratio results (9) for h_{Che} and (12) for h_{CT} apply to the extended object hit ratio without change. The byte hit ratio for both approaches is given by:

$$h_{\mathrm{Che,Byte}}=\sum_{k=1}^{N} s_k p_k(1-e^{-p_k \Delta_{\mathrm{LRU}}})/\sum_{k=1}^{N} s_k p_k; \quad (16)$$

$$h_{\mathrm{CT,Byte}}=\sum_{k=1}^{N} s_k p_k(1-(1-p_k)^{\overline{CT}_{IRM}^{LRU}})/\sum_{k=1}^{N} s_k p_k. \quad (17)$$

A. Imprecise Approximation Cases for Small Cache Size

A simple small cache example with $N = 2$ objects of size $s_1 = s_2 = 10$ and IRM request probabilities $p_1 = p_2 = 0.5$ leads to $h_{\mathrm{Che}}(M) = h_{\mathrm{CT}}(M) = M/20$ for $M \le 20$, which largely deviates from $h_{\mathrm{LRU}} = 0$ for $M < 10$, $h_{\mathrm{LRU}} = 0.5$ for $10 \le M < 20$. The deviations Δh_{Che}, Δh_{CT} are ramping up to 45% for $M = 9$ and $M = 19$. The example suggests the following improvements of the basic h_{Che} and h_{CT} approach to reduce such deviations:

- Oversize correction
  We ignore objects, which do not fit into the cache, such that $\forall k$: $s_k \le M$. In this way, $h_{\mathrm{Che}}(M) = h_{\mathrm{CT}}(M) = 0$ is corrected in the range $M < 10$ in this example.
- Unused cache space (*UCS*) correction
  We compute $h_{\mathrm{Che}}(M^*)$ and $h_{\mathrm{CT}}(M^*)$ for reduced cache size M^*, where the mean amount of unused cache space $E[UCS(M)]$ is subtracted $M^* = M - E[UCS(M)]$. In the example, the *UCS* correction eliminates the deviations on the entire range $0 \le M \le 20$ of the hit ratio curve (HRC).

B. Mean UCS Computation Model and Algorithm

The oversize correction is a generally simple task, whereas the unused cache space is changing as a dynamic stochastic process depending on the cache content. Therefore, it requires more elaborate modelling to determine $E[UCS(M)]$.

We evaluate the mean *UCS* by a simplified model that follows *UCS* changes caused by LRU replacement of objects per request and captures the steady state *UCS* behaviour.

Let s_m^R denote the size of the object that is referenced in the m^{th} request and $s_{m,k}^E$ the sizes of the eviction candidates. The cache fill level before the m^{th} request is denoted as $F_m \le M$. It remains unchanged in case of a cache hit, but F_m is modified after each cache miss by insertion of the requested object and compensating evictions from the LRU cache:

$$F_{m+1} = F_m + s_m^R - \sum_{k=1}^{K} s_{m,k}^E.$$

The number K of required evictions follows the conditions:

$$K = 0 \quad \text{if } F_m + s_m^R \le M \text{ or otherwise}$$

$$F_m + s_m^R - \sum_{k=1}^{K} s_{m,k}^E \le M < F_m + s_m^R - \sum_{k=1}^{K-1} s_{m,k}^E.$$

We iteratively compute F_m in the format of discrete distributions $f_m(j) = \mathrm{Prob}(F_m = j)$ until steady state conditions are approached and a stabilized mean unused cache space is obtained: $\mathrm{E}[UCS(M)] = M - \lim_{m\to\infty} E[F_m]$.

In order to determine which object enters the cache after a cache miss, we refer to Che's approximation for providing an estimate $e^{-p_k \Delta_{\mathrm{LRU}}}$ of the probability that object O_k is outside of the cache. Then the probability that O_k is the next object to enter the cache is given by

$$p_{Enter}(O_k) \approx p_k e^{-p_k \Delta_{\mathrm{LRU}}} / \Sigma_k p_k e^{-p_k \Delta_{\mathrm{LRU}}}.$$

Consequently, the distribution of the size s_m^R of an object entering the cache is given by $s(j) = \Sigma_k p_{Enter}(O_k)\ \mathrm{Prob}(s_k = j)$.

The distribution $s(j)$ also characterizes the size of evicted objects because the frequency of evictions of an object is the same as the frequency for (re-)entering the cache. We finally assume that all sizes s_m^R and $s_{m,k}^E$ of inserted and evicted objects are independent and follow the same distribution $s(j)$ and we denote the maximum object size as $s_{\max}$.

When the fill level F_m is in the range $M - s_{\max} < F_m \le M$ then $F^*_m = F_m + s_m^R$ is bounded by $M - s_{\max} < F^*_m \le M + s_{\max}$. The evictions reduce the fill level again into $M - s_{\max} < F_{m+1} \le M$. We start with F_1 at the maximum achievable fill level, which is equal to M or at least in the range $M - s_{\max} < F_1 \le M$. An iteration step proceeds from $f_m(j) = \mathrm{Prob}\{F_m = j\}$ for $M - s_{\max} < j \le M$ via a convolution for inserting an object with size distribution $s(k)$ towards $f_m^*(j) = \mathrm{Prob}\{F_m + s_m^R = j\}$.

Thereafter, the convolution result $f_m^*(j)$ is transformed into the next fill level distribution $f_{m+1}(j)$ corresponding to a required number of evictions via the following C++ pseudocode:

```
for (k = M + s_max; k > M; k--)
    for (j = 1; j ≤ s_max; j++)  f_m*(k − j) += f_m*(k)·s(j);
for (j = 0; j < s_max; j++)      f_{m+1}(M − j) := f_m*(M − j);
```

Each iteration step to determine $f_{m+1}(k)$ from $f_m(k)$ has computational complexity $O(s_{max}^2)$. The entire computation of $E[UCS(M)] = M - \lim_{m\to\infty} E[F_m]$ has complexity $O(s_{max}^2)$, where we experience a fast convergence to a steady state. For $s_{\max} \le 10000$, the $E[UCS(M)]$ computation stays below 1s on a usual PC. Even for a broader range of object sizes, we can keep the distribution $s(j)$ within 10000 steps via coarser discretization.

C. Effect of the USC Correction on the Precision

Finally, we demonstrate the effect of the oversize and *UCS* correction on the precision of Che's approximation in a scenario with objects of largely varying size. Therefore, the object catalogue includes 50 objects of unit size $s_1 = \ldots = s_{50} = 1$ and request probabilities $p_1 = \ldots = p_{50} = 1\%$, as well as 5 objects with 10 - 50-fold size $s_{50+k} = 10k$ and $p_{50+k} = 10\%$ for $k = 1, \ldots, 5$.

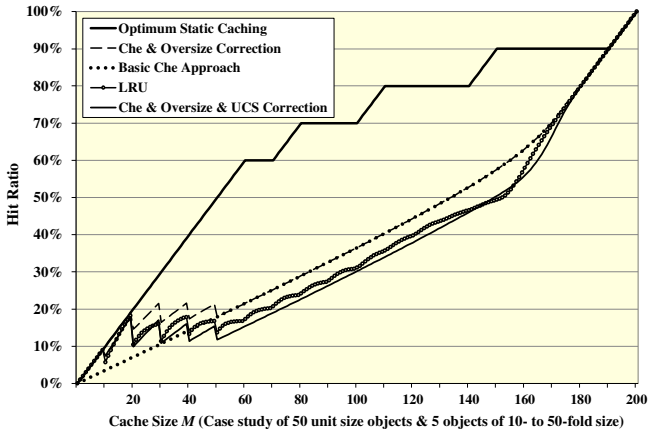

Figure 4: Effect of oversize exclusion and *UCS* correction

Figure 4 shows hit ratio curves (HRCs) for

(1) optimum static caching due to a knapsack solution, which prefers objects with maximum ratio p_k/s_k,

(2) LRU via simulative evaluation,

(3) the basic Che approach according to (15) [12],

(4) Che's approach without oversize objects for $M < 50$, and

(5) Che's approach with oversize and mean *UCS* correction.

A zigzag shape of the LRU HRC is apparent in the cache size range $9 \le M \le 50$. For $M = 9$, the LRU cache is filled with 9 out of the 50 unit size objects, while the larger objects do not fit, yielding a hit ratio $h_{\mathrm{LRU}}(9) = 9\%$. For $M = 10$, object O_{51} of size 10 can enter and fill the cache, still leading to 10% hit ratio at this stage. However, a next request to a unit size object will replace O_{51} by a single object of size 1 in an almost empty cache. This effect reduces the LRU HRC from $h_{\mathrm{LRU}}(9) = 9\%$ downto $h_{\mathrm{LRU}}(10) \approx 5.7\%$, despite of increasing cache size. The LRU HRC is staggering several times, when more large objects can enter, until a last step down from $h_{\mathrm{LRU}}(49) \approx 16.8\%$ downto $h_{\mathrm{LRU}}(50) \approx 13.7\%$.

On the other hand, the basic Che HRC is monotonously increasing. Oversize correction is required in order to follow the zigzag shape of the LRU HRC, but leads to overestimation. The HRC for combined oversize and *UCS* correction comes significantly closer to the LRU HRC.

As the maximum absolute deviation $\Delta_{\max} = \max(|\Delta h_{\mathrm{Che}}|)$ and the standard deviation σ we obtain

- for the basic Che approach: $\Delta_{\max} \approx 11.45\%; \sigma \approx 4.53\%;$
- without oversize objects: $\Delta_{\max} \approx 8.56\%; \sigma \approx 4.17\%;$
- with oversize & *UCS* correction: $\Delta_{\max} \approx 3.7\%; \sigma \approx 1.74\%.$

In general, oversize and unused cache space corrections are relevant for the first part of the HRC with small M, but will become negligible for large N, M, if the cache size is much larger than the size of single objects.

As a final remark, it is obvious from this and many other studies [1][10][22][24] that LRU caching performance can be poor for IRM and moderately correlated request pattern, when compared to optimized score based caching methods, especially when the variance of the object sizes is high. This is crucial for web caching, where the size of cacheable data units is scattering over a wide range from kByte to Mbyte and beyond [1][3]. Zigzag-shaped HRC curves generally indicate performance deficits of a caching method, because optimized strategies can at least preserve the hit ratio level, when the cache size is increasing.

CONCLUSIONS

Since a cache filling phase is sufficient for the convergence time (CT) of LRU to steady state IRM behavior, the CT analysis can be restricted to a cache startup phase, whose hit ratio development is still characterized by the classical LRU analysis [18]. Most other caching strategies show the same behavior as LRU in cache filling phases, followed by a second transient phase from LRU level to their own steady state hit ratio level. The mean LRU CT is approximated in approaches by Fagin [11] as "*window size T*" to determine the "*expected working set miss ratio*" and by Che et al. [8] as "*characteristic time*".

We complement results [4][5][12][26] confirming asymptotic accuracy of both approximations [8][11] of the LRU hit ratio for large caches with a quantitative study covering large deviation cases. Request distributions with maximum absolute deviation of $|\Delta h_{\mathrm{Che}}| \approx 8.25\%$ are identified for Che's approach [8] and with $|\Delta h_{\mathrm{CT}}| \approx 5.2\%$ for the CT approach [11]. The exact LRU analysis [18] is tractable for small caches and offers an alternative especially in the range, where the approximations are subject to large errors.

On the other hand, our results strongly suggest monotonously decreasing deviations with the cache size M and confirm generally good accuracy for usual cache sizes with $|\Delta h_{\mathrm{Che}}| < 1\%$ and $|\Delta h_{\mathrm{CT}}| < 1.3\%$ already for $M \ge 10$. However, our quantitative study is focused on caches of limited size, leaving a general proof of those results for future study.

For LRU caches with objects of different size, we derive an extension of the exact hit ratio formula [18]. Based on the Che and CT approaches, an improved approximation scheme is provided, which takes the fraction of unused cache space into account. This leads to significantly better precision especially for small caches and when the variance of the size of requested objects is high. Otherwise, if the object sizes are much smaller than the cache size, we expect a statistical multiplexing effect that leads to asymptotic convergence of the extended Che and CT estimates for varying object sizes, similar to already proven properties for unit size objects.

REFERENCES

[1] M.F. Arlitt and C.L. Williamson, Internet web servers: Workload characterization and performance implications, IEEE Trans. on Networking 5/5 (1997) 631-645

[2] H. Ben-Ammar et al., On the performance analysis of distributed caching systems using a customizable Markov chain model, Journal of Network and Computer Appl. 130, Elsevier (2019) 39-51

[3] D.S. Berger, R.K. Sitaraman and M. Harchol-Balter, AdaptSize: Orchestrating the Hot Object Memory Cache in a Content Delivery Network, Proc. 14th USENIX Symposium NSDI (2017) 483-498

[4] C. Berthet, Approximation of LRU caches miss rate: Application to power-law popularities, arXiv:1705.10738 (2017) 1-36

[5] M. Brenner, A Lyapunov analysis of LRU, Master thesis, Univ. of Illinois (2020) 1-42

[6] A.K. Bhide, A. Dan, and D.M. Dias, A simple analysis of the LRU buffer policy and its relationship to buffer warm-up transient, IEEE Proc. on the 9th conference on data engineering, Vienna, Austria (1993) 125-133

[7] L. Breslau et al., Web caching and Zipf-like distributions: Evidence and implications, Proc. IEEE Infocom (1999) 126-134

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

[9] A. Dan and D. Towsley, An approximate analysis of the LRU and FIFO buffer replacement schemes, Proc. ACM SIGMETRICS, Boulder, Colorado, USA (1990) 143-152

[10] H. ElAarag, Web proxy cache strategies: Simulation, implementation and performance evaluation, Springer Publ. (2013) 1-103

[11] R.Fagin, Asymptotic miss ratios over independent references, Journal of Computer and System Sciences 14 (1977) 222-250

[12] C. Fricker, P. Robert, J. Roberts, A versatile, accurate approximation for LRU cache performance, Proc. ITC 24, Krakow, Poland (2012) 1-8

[13] E. Gelenbe, A unified approach to the evaluation of a class of replacement algorithms, IEEE Trans. on Comp., 22/6 (1973) 611-618

[14] G. Hasslinger et al., Performance evaluation for new web caching strategies combining LRU with score-based selection, Computer Networks 125 (2017) 172-186

[15] G. Hasslinger et al., Web caching evaluation for Wikipedia request statistics, Proc. IEEE WiOpt Symposium, Paris, France (2017) 1-6

[16] G. Hasslinger et al., Fast and efficient web caching methods regarding the properties per data, Proc. IEEE CAMAD, Limassol, Cyprus (2019) 1-7

[17] B. Jiang, P. Nain, and D. Towsley, On the Convergence of the TTL Approximation for an LRU Cache under Independent Stationary Request Processes, ACM TOMPECS 3/4-20 (2018) 1-31

[18] W.F. King III, Analysis of demand paging algorithms, Proc. IFIP Congress, Ljubljana, Yugoslavia (1971) 485-490

[19] N. Laoutaris H. Che, I. Stavrakakis, The LCD interconnection of LRU caches and its analysis, Performance Evaluation 63/7 (2006) 609-634

[20] J. Li, S. Shakkottai, J.C.S. Lui and V. Subramanian, Accurate learning or fast mixing? Dynamic adaptability of caching algorithms, IEEE JSAC 36/6 (2018) 1314-1330

[21] J. McCabe, On serial files with relocatable records, Operations Research 13/4 (1965) 609-618

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

[23] K. Ntougias et al., Coordinated caching and QoS-aware resource allocation for spectrum sharing. Wireless Personal Comm. 112 (2020)

[24] G.S. Paschos, G. Iosifidis and G. Caire, Cache optimization models and algorithms, Foundations and Trends in Communications and Information Theory 16/3-4 (2020) 156-345

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

[26] P. Poojary et al., A coupon collector based approximation for LRU cache hits for Zipf requests, IEEE Proc. IFIP WiOpt Symp. (2021) 1-8

[27] S. Traverso et al., Unraveling the impact of temporal and geographical locality in caching systems, IEEE Trans. Multimedia (2015) 1839-54

[28] A.K.Y. Wong et al., Exact transient analysis on LRU cache startup for IoT, Proc. ACM Conf. on Inform. Tech.: IoT & Smart City (2021) 310-315