future internet

MDPI

Article

Scope and Accuracy of Analytic and Approximate Results for FIFO, Clock-Based and LRU Caching Performance

Gerhard Hasslinger [1,*], Konstantinos Ntougias [2], Frank Hasslinger [3] and Oliver Hohlfeld [4]

[1] Deutsche Telekom, 64289 Darmstadt, Germany
[2] Department of Electrical and Computer Engineering, University of Cyprus, Nicosia 22006, Cyprus
[3] Department of Computer Science, Darmstadt University of Technology, 64289 Darmstadt, Germany
[4] Distributed Systems Group, University of Kassel, 34127 Kassel, Germany
* Correspondence: hasslinger@informatik.tu-darmstadt.de

Abstract: We evaluate analysis results and approximations for the performance of basic caching methods, assuming independent requests. Compared with simulative evaluations, the analysis results are accurate, but their computation is tractable only within a limited scope. We compare the scalability of analytical FIFO and LRU solutions including extensions for multisegment caches and for caches with data of varying sizes. On the other hand, approximations have been proposed for the FIFO and LRU hit ratio. They are simple and scalable, but their accuracy is confirmed mainly through asymptotic behaviour only for large caches. We derive bounds on the approximation errors in a detailed worst-case study with a focus on small caches. The approximations are extended to data of different sizes. Then a fraction of unused cache space can add to the deviations, which is estimated in order to improve the solution.

Keywords: FIFO; RANDOM; LRU; LFU; clock-based and multisegment caches; Markov analysis; hit ratio approximations; deviation bounds; variable data size

Citation: Hasslinger, G.; Ntougias, K.; Hasslinger, F.; Hohlfeld, O. Scope and Accuracy of Analytic and Approximate Results for FIFO, Clock-Based and LRU Caching Performance. *Future Internet* **2023**, *15*, 91. https://doi.org/10.3390/fi15030091

Academic Editors: Sachin Sharma and Nouman Ashraf

Received: 6 January 2023
Revised: 12 February 2023
Accepted: 13 February 2023
Published: 24 February 2023

1. Introduction: Basic Caching Methods and Their Evaluations

Caching strategies are essential for the performance of local caches in CPU, GPU and database systems [1] as well as for content delivery on the Internet [2,3]. A set of basic caching strategies, such as first-in-first-out (FIFO), least recently/frequently used (LRU/LFU) and clock-based caching [1,4–9], were studied in early caching approaches almost 50 years ago, and they are still preferably applied today. Other caching methods that include detailed properties, caching costs and values per data object can optimize the efficiency on the basis of score rankings [10,11] or machine-learning approaches [12,13]. However, FIFO and LRU are still attractive caching principles because of their simple and fast data management [3,14–16].

We compare analysis and approximation methods for the FIFO and LRU hit ratio evaluation, where RANDOM and clock-based caches are included, because their performance is partly equivalent to FIFO [6]. Moreover, simulation is a usual performance evaluation approach for caches. Simulations are flexibly applied for many caching methods and variants [2,11,17–19], but they are subject to random deviations, which require long runs for accurate results in narrow confidence intervals.

Exact analytic hit ratio solutions were derived long ago by W.F. King [8] and E. Gelenbe [6] for the independent reference model. In the past decade, the state of the art has advanced towards solutions for multisegment/multilevel caches [20], and for LRU caches with data objects of different sizes, including the cache startup or filling phase [21,22]. However, those exact solutions are tractable only for small caches.

In addition to exact analysis and simulation, approximation formulas form a third pillar for a simpler evaluation of basic caching methods, as proposed by Fagin [23] and

Che et al. [24] for LRU as well as by Dan and Towsley [25] for FIFO. Recent research has approved asymptotical exactness of both LRU variants for large caches [26–29].

The motivation and purpose of this work is to present a complete picture of the currently available analysis and approximation results for FIFO and LRU. Our focus is mainly on FIFO results [14,16], which are only sparsely addressed in the literature, in contrast to LRU [9,18,21,22]. The product form solution of the FIFO hit ratio for independent requests [6,8] is scalable for large caches and applicable to clock-based caching (CpR: clock per request), in contrast to LRU analysis results. We show that the FIFO approximation [25] has similar properties, as confirmed for LRU [26–29] for data of unit size, and we extend the approaches for data of variable sizes as required for web content delivery.

An overview of the similarities and differences in the results for FIFO and LRU is provided in Table 1. The scope of the analysis methods is clarified in the main part, and the accuracy of the approximations is studied in an extensive quantitative investigation. On the basis of detailed insights and a comparison of the alternative cache performance evaluation methods, we finally recommend the most appropriate evaluation options for the considered caching use cases.

Table 1. Overview of analytical and approximate cache performance results.

Solution Type and References		Applies to	Scalable Computation	Different Object Sizes	Maximum Deviations
Analytic Cache Hit Ratio Results	Product Form [8] Equations (1) and (2)	FIFO, Clock p.R., RANDOM	☑ [30] Section 2.2	☒ No Common Solution fo FIFO, CpR, RANDOM	Numerially Exact Evaluations
	Extension [20] Equation (10)	Multi-Level Caches	Complex, but Scalable		
	LRU Formula [8] Equations (11) and (12)	LRU & Cache Fill Phases	☒ Only for Small Caches	☑ Extension Equation (13) [21]	
Approximations	Dan et al. [25] Equations (4) and (5)	FIFO, CpR, RANDOM	☑	☑ but less accurate Section 6.3, Section 6.4, Section 6.5	16% for M = 1 <3% for M $\geq$ 10
	Fagin [23] Equation (8)	LRU & Cache Fill Phases			5.4% for M = 2 <1.3% for M $\geq$ 10
	Che et al. [24] Equation (7)				8.5% for M = 1 <1% for M $\geq$ 10

As new contributions of this work,

- product form solutions are extended to clock-based methods and combinations of those methods, with FIFO and RANDOM for single and multilevel caches,
- the scope of analytical solutions is clarified regarding scalable computation as well as the solutions' applicability for data of different sizes, and
- quantitative evaluations of the accuracy of approximations identify the worst cases and error bound extensions for varying data sizes in caches.

We start with exact analysis results for the FIFO, CpR and RANDOM hit ratio and assess their scalability, in Section 2. The precision of approximations for FIFO and LRU caching strategies is evaluated in Sections 3 and 4. Extensions to multisegment caches are addressed in Section 5. Section 6 focuses on FIFO solutions for objects of different sizes in comparison to LRU. Section 7 summarizes the main results and their limitations.

2. Markov Analysis Results for Basic Caching Strategies

W.F. King 8 provided steady state FIFO and LRU hit ratio formulas, assuming a limited cache size for M objects of unit sizes and an independent reference model (IRM). The latter is characterized by a fixed catalogue of N objects $O_1, \ldots, O_N$, which are referenced with probabilities $p_1, \ldots, p_N$ in each request. The IRM request pattern with the Zipf-distributed popularity of the data has been confirmed manifold as a realistic model for web request traces [17,31,32]. Moderate correlation among requests and changes in the working set of relevant data are also relevant [31–33].

2.1. Common Hit Ratio Analysis of FIFO, RANDOM and Clock-Based Caching

FIFO caches replace the object that is present in the cache for the longest time with the requested object in the case of a cache miss. W.F. King [8] showed that FIFO caching follows a Markov process under IRM with a transition per request, whose steady state content distribution has a product form solution. The probability $p_{FIFO}(O_{k1}, \cdots, O_{kM})$ to find the objects $O_{k1}, \cdots, O_{kM}$ in positions $1, \ldots, M$ of a FIFO cache is given by

$$p_{FIFO}(O_{k1}, \cdots, O_{kM}) = c\, p_{k1} \cdot \cdots \cdot p_{kM} \ (\forall\, j \neq l\text{: } O_{kj} \neq O_{kl};\ M\text{: cache size; } c\text{: norm. constant}).$$

Gelenbe [6] proved the same product form solution to be valid also for a RANDOM caching strategy, which randomly chooses the eviction candidate with probability $1/M$ among the objects in the cache. Moreover, we confirm a broader scope of product form solutions, including clock-based caching and combinations of FIFO/RANDOM/clock.

Clock methods indicate an eviction candidate by a clock hand, which steps one position forward per request (CpR). Upon a cache miss, the requested object replaces the eviction candidate. Corbato [4] proposed several variants of the basic CpR method. Score-gated clock [11] is a clock scheme which admits new content only if it has a higher score than the eviction candidate. The steady state probabilities $p_{CpR}(O_{k1}, \cdots, O_{kM})$ of CpR cache content follow the same product form as for FIFO and RANDOM:

$$p_{CpR}(O_{k1}, \cdots, O_{kM}) = p_{FIFO}(O_{k1}, \cdots, O_{kM}) = p_{RANDOM}(O_{k1}, \cdots, O_{kM}) = c\, p_{k1} \cdot \cdots \cdot p_{kM} \qquad (1)$$

As the main step of a proof, we insert this solution into the equilibrium equations of the underlying Markov chain for CpR cache content. Figure 1 illustrates which transitions are possible from previous states to the content $O_{k1}, \cdots, O_{kM}$.

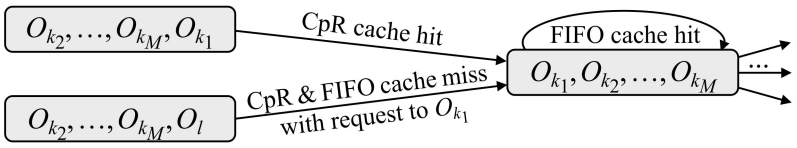

Figure 1. State transitions for CpR and FIFO to the cache content $O_{k1}, \cdots, O_{kM}$.

For a cache hit, a cyclic clock shift leads from content $O_{k2}, \cdots, O_{kM}, O_{k1}$ to $O_{k1}, \cdots, O_{kM}$ with a corresponding transition probability $p_{k1} + \cdots + p_{kM}$. Otherwise, a cache miss leads to $O_{k1}, \cdots, O_{kM}$if O_{k1} is requested as an external object, i.e., with transition probability p_{k1} from a previous content state $O_{k2}, \cdots, O_{kM}, O_l$. We obtain the following equilibrium equations corresponding to one-step transitions per request to $O_{k1}, \cdots, O_{kM}$:

$$p_{CpR}(O_{k_1}, \ldots, O_{k_M}) = p_{CpR}(O_{k_2}, \ldots, O_{k_M}, O_{k_1}) \sum_{j=1}^{M} p_{k_j} + p_{k_1} \sum_{\substack{l=1 \\ l \notin \{k_1, \ldots k_M\}}}^{N} p_{CpR}(O_{k_2}, \ldots, O_{k_M}, O_l)$$

The product form solution is substituted and verified for those equilibrium equations:

$$cp_{k_1} \ldots p_{k_M} = cp_{k_2} \ldots p_{k_M} p_{k_1} \sum_{j=1}^{M} p_{k_j} + p_{k_1} \sum_{\substack{l=1 \\ l \notin \{k_1, \ldots k_M\}}}^{N} cp_{k_2} \ldots p_{k_M} p_l \Leftrightarrow$$

$$1 = \sum_{j=1}^{M} p_{k_j} + p_{k_1} \sum_{\substack{l=1 \\ l \notin \{k_1, \ldots k_M\}}}^{N} \frac{p_l}{p_{k_1}} = \sum_{j=1}^{M} p_{k_j} + \sum_{l=1}^{N} p_l - \sum_{l=1}^{M} p_{k_l} = 1$$

This main step for proving the CpR steady state solution applies to FIFO and RANDOM strategies in a similar way [6,8]. Transitions in steady state are encountered only

between content states representing a full cache, while states with less than M objects are transient. The proof of the product form solution, Equation (1), is completed by checking whether the Markov chain is ergodic; i.e., a transition path must lead from each content state $O_{k1}, \ldots, O_{kM}$ to each other with nonzero probability. The underlying IRM Markov processes for the RANDOM and CpR strategies are generally ergodic if $\forall k$: $p_k > 0$, whereas FIFO has a nonergodic exception if and only if $N = M + 1$. These results are covered by general criteria for ergodic caching networks derived by Rosensweig et al. [34].

Finally, the product form solution implies a common steady state hit ratio for the next request as the sum of request probabilities of all data objects stored in the cache [6,8]:

$$h_{IRM}^{FIFO} = h_{IRM}^{CpR} = h_{IRM}^{RANDOM} = \sum_{k_1,k_2,\ldots k_M=1}^{N} p_{k_1} p_{k_2} \cdots p_{k_M} \sum_{j=1}^{M} p_{k_j} / \sum_{k_1,k_2,\ldots k_M=1}^{N} p_{k_1} p_{k_2} \cdots p_{k_M} \quad (2)$$

Moreover, the common solution extends to a broader class of variants and combined schemes. For example, when we alternate between FIFO, RANDOM and CpR in subsequent requests either in a periodic pattern or in a randomized pattern, the previous core verification step still applies per request. Variants of the clock-hand movement by skipping several objects per requests again leave the steady state solution unchanged, provided that the Markov process is still ergodic [34]. The FIFO product form solution has been derived in alternative ways via reversibility properties of the underlying Markov process by Marin et al. [35] and via fluid flow approximation by Tsukada et al. [36].

2.2. Scalable Iterative Evaluation of the Product Form Solution

An iterative evaluation of the product form (2) with scalable complexity $O(MN)$ was proposed by Fagin and Price [30], which makes the FIFO hit ratio computations tractable even for large caches, in contrast to the LRU analysis formula of Equations (11)–(13). Equation (2) is evaluated in an extended format of intermediate results $s(m, n)$ and $h(m, n)$:

$$s(m,n) = \sum_{k_1,k_2,\ldots k_m=1}^{n} p_{k_1} \ldots p_{k_m} \forall m = 1, \ldots, M \text{ and } \forall\, n = m, \ldots,\ N; \quad (3)$$

$$h(m,n) = \sum_{k_1,k_2,\ldots k_m=1}^{n} p_{k_1} \ldots p_{k_m} \sum_{l=1}^{m} p_{k_l};\ h_{IRM}^{FIFO} = h(M,N)/s(M,N)$$

Next, the following scheme can be used to compute h_{IRM}^{FIFO}, etc. via Equation (3) [30]:

$$s(m,\, n+1) = s(m,\, n) + s(m-1,\, n)\, p_{n+1} \text{ with initialization } s(0,\, n) = p_1 + \ldots + p_n\ ;$$

$$h(m, n+1) = h(m,\ n) + h(m-1,\ n)\ p_{n+1} + s(m-1,\ n) p_{n+1}^2; \text{and} h(0,\ n) = p_1^2 + \ldots + p_n^2.$$

However, a direct implementation of this scheme is subject to numerical instability for large m, n, because the components $s(m, n)$, $h(m, n)$ of Equation (3) are partly extremely small, in a range below 10^{-100}, 10^{-1000}, etc. and thus below the usual CPU number representation. In order to improve numerical accuracy for large m, n, we sort the objects according to their request probabilities $p_1 \leq p_2 \leq \ldots \leq p_n$ and start summations with the smallest values. Moreover, we add a factor 10^{-20k} ($k \in N$) to the real number representations of $s(m, n)$ and $h(m, n)$, such that the representation of y in the range $(10^{-20(ky+1)}, 10^{-20ky})$ has an integer component $k_y \in N$ and a real component $r_y \in (10^{-20}, 1)$, such that $y = r_y \cdot 10^{-20ky}$. Thereafter, 10^{-1000} is included for $(r_y\ , k_y) = (1, 50)$, etc. In this way, we finally achieve an accurate and scalable computation of the product form solution, as confirmed by a comparison with the simulation results of numerous examples, including the results in Figure 2 and Table 2.

3. Approximation of the IRM Hit Ratio for FIFO

Because the analytical FIFO solution of Equations (2) and (3) needs an extended number representation for large caches and the LRU solution of Equations (11)–(13) is not scalable, simpler approximations have been proposed. The FIFO hit ratio approximation refers

to the time spans that an object O_k spends inside the cache ($T_{InCache}$) and outside ($T_{Extern,k}$), which are measured by the number of requests in both time spans. For independent requests, the external phase $T_{Extern,k}$ is geometrically distributed with mean $\overline{T}_{Extern,k} = 1/p_k$ until the next request to O_k.

When an object enters the cache, it starts from the top FIFO position and steps down by one place per cache miss. We assume a constant cache miss probability $1 - h_{IRM}^{FIFO}$, independent of the current cache content. Next, the mean number of requests that an object spends in a FIFO cache is the same for all objects: $\overline{T}_{InCache} \approx M/(1 - h_{IRM}^{FIFO})$. Finally, we obtain the hit ratio per object O_k: $h_k^{FIFO} = p_k \overline{T}_{InCache} / (\overline{T}_{InCache} + \overline{T}_{Extern,k})$. In total,

$$h_{IRM}^{FIFO} = \sum_{k=1}^{N} h_k^{FIFO} = \sum_{k=1}^{N} p_k \frac{\overline{T}_{InCache}}{\overline{T}_{InCache} + \overline{T}_{Extern,k}} = \sum_{k=1}^{N} \frac{p_k \overline{T}_{InCache}}{\overline{T}_{InCache} + 1/p_k} \quad (4)$$

This leads to the following iterative computation scheme for $\overline{T}_{InCache}$:

$$\overline{T}_{InCache} = M/(1 - h_{IRM}^{FIFO}) = M/(\sum_{k=1}^{N} p_k - \frac{p_k \overline{T}_{InCache}}{\overline{T}_{InCache} + 1/p_k}) = M/\sum_{k=1}^{N} \frac{1}{\overline{T}_{InCache} + 1/p_k} \quad (5)$$

Starting from $h_{IRM}^{FIFO} = 0 \Rightarrow \overline{T}_{InCache} = M$, $\overline{T}_{InCache}$ is monotonously increasing in each iteration step of Equation (5). There is a unique solution, and the iteration is converging towards this solution. The computation effort of Equation (5) is $O(N)$ per iteration round.

The FIFO (RANDOM, CpR) hit ratio approximation of Equation (5) is equivalent to the approach by Dan and Towsley [25]. Based on a slightly more restrictive assumption of Poisson request processes per object, Garetto et al. [37] provide an alternative derivation by using queueing theory and Dehghan et al. [10] by using time-to-live caching.

3.1. Precision of the FIFO Approximation for Zipf-Distributed IRM Requests

We compare the exact FIFO hit ratio analysis with the approximation for cases of Zipf-distributed IRM requests [31] with $p_k = \alpha\, k^{-\beta}$ for $k = 1, \ldots, N$; $\alpha = 1/\Sigma_k\, k^{-\beta}$. Zipf-distributed request pattern has been reported manifold in content distribution on the Internet for the popularity of files, videos, etc. [31–33]. Figure 2 shows results for $\beta = 1$ and for varying object catalogue sizes $N = 10, \ldots, 10^6$, with cache sizes M over the entire range $[1, \ldots, N]$. The largest deviations of up to 1.7% are encountered for the smallest M, N. For large N, M, the deviations are often tiny, going down to the range of 10^{-6}–10^{-7}, where asymptotical exactness can be expected due to a statistical multiplexing effect. As a general trend, we experience the deviations to decrease with the variance of the popularity distribution. Thus, for independent Zipf-distributed requests, the deviations are increasingwith the shape parameter β and decreasing with M, N.

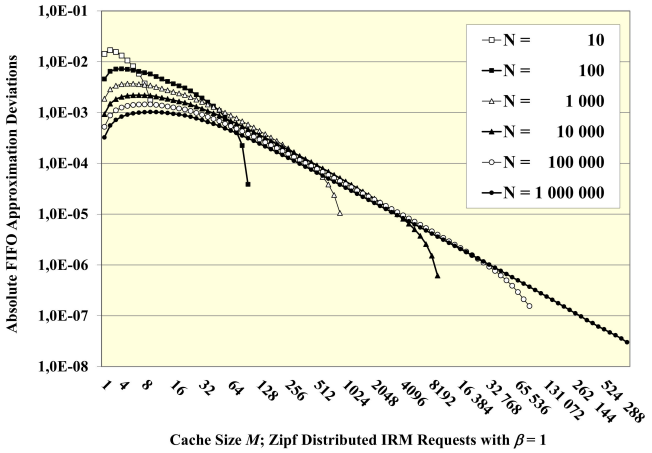

Figure 2. Deviations of FIFO approximations of Equations (4) and (5) from exact results of Equations (2) and (3).

3.2. Maximum Error Cases of the FIFO Approximation for Small Cache Size M

The FIFO hit ratio approximation is highly accurate for large caches, but larger errors are encountered for small caches. In order to identify maximum error cases, the following approach aims to cover the entire range of IRM popularity distributions for small M, N.

In particular, we checked all 204 266 popularity distributions whose request probabilities are multiples of 1/50 = 2%, such that $p_k = i_k/50$ with integers i_k for $k \leq N \leq 50$. The approximation of Equations (4) and (5) is compared to the product form analysis of Equation (2) for all those distributions and for all cache sizes $M < N$, yielding over a million deviation results. The computation required 2 days on a usual PC, but an exponential increase in the number of distributions will soon impede extensions to a finer raster of 1/51, 1/52, etc.

We separated the results for each cache size M and sorted them according to decreasing deviations. Finally, we obtained lists of the top-K largest deviations of the FIFO approximation for all popularity distributions in the 2% raster. They strongly suggest that maximum deviations are encountered for popularity distributions of the following type:

$$p_1 = p_2 = \cdots = p_n = p/n;\ p_{n+1} = \cdots = p_N = (1-p)/(N-n) \to 0 \text{ for } N \to \infty, \tag{6}$$

i.e., there are n data objects with the same popularity among many more objects with negligible request probabilities. The parameters n, p specify extreme distributions of the type (6) for different cache sizes M. In our study of popularity distributions in the 2% raster, a sorted list of the largest deviations starts with hundreds of cases which are closest to a specific case of the format of Equation (6).

The same behaviour has also been experienced by Fagin [23] and Che et al. [24] for maximum deviations of approximations to the LRU hit ratio; see Section 4. An example of the top-10 maximum error cases of approximations in the 2% raster is shown in Figure 4, which are all near the corresponding maximum error case of the format (6).

The distribution type (6) leads to simple direct solutions for the approximation Equation (5) and for the exact approach of Equation (2). A substitution of Equation (6) into Equations (4) and (5) results in a quadratic equation for $\overline{T}_{InCache}$ and an explicit solution for h_{IRM}^{FIFO}:

$$h_{IRM}^{FIFO} \approx n(p/n)\overline{T}_{InCache}/(\overline{T}_{InCache} + n/p);\ h_{IRM}^{FIFO} \approx 1 - M/\overline{T}_{InCache} \Rightarrow$$

$$\overline{T}_{InCache} \approx M(\overline{T}_{InCache} + n/p)/(\overline{T}_{InCache} + n/p - p\overline{T}_{InCache});\Rightarrow$$
$$\overline{T}_{InCache} \approx (M - n/p \pm \sqrt{(M+n/p)^2 - 4Mn})/(2-2p)$$

The special format of Equation (6) also allows for computing the probabilities q_m that we find m out of n popular objects in the cache for $m = 0, \ldots, \min(n, M)$. In this way, we obtain the exact hit ratio result $h_{IRM}^{FIFO} = \Sigma_m\, q_m \cdot m \cdot p/n$, especially for the format (6). We derive q_m by collecting all the corresponding components from Equation (2):

$$q_m = c \binom{n}{m} \binom{N-n}{M-m} \left(\frac{p}{n}\right)^m \left(\frac{1-p}{N-n}\right)^{N-m}$$

A further evaluation of this expression in the limit $N \to \infty$ yields the following simple scheme:

$$q_{m+1}/q_m = (n-m)(M-m)p/[n(m+1)(1-p)],$$

which leads to a direct evaluation of $q_0, \cdots, q_M$ and finally of h_{IRM}^{FIFO}.

In Table 2, the worst FIFO error cases are listed for $M = 1, \ldots, 10$. The parameters p, n specify the probabilities $p_1 = \cdots = p_n = p/n$ of the extreme format (6). The approximation is exact for uniform distributions, as confirmed by substituting $h_{IRM}^{FIFO} = M/N$ and $p_k = 1/N$ into Equation (5). Moreover, we observe without proof that the errors are always negative, i.e.,

the approximation of Equations (4) and (5) seems to establish a lower bound. The main conclusions from Table 2 on the accuracy of the FIFO approximation are as follows:

- Considerable errors of up to 16.5% are encountered for small cache size M.
- They are reduced towards a fair accuracy with errors below 3% for $M \geq 10$.

Table 2. Maximum deviations of FIFO approximations of Equations (4) and (5) for cache size M.

Maximum Deviations of the FIFO, RANDOM & CpR Approximation for Cache Size M: Worst Cases Are Request Distributions of the Type (6) with $n = M$				
M	$p_1 = \cdots = p_n$ $(n = M)$	Exact Result h_{IRM}^{FIFO} of Equation (2)	Approximation of Equations (4) and (5)	Maximum Deviation
1	0.8838	78.11%	61.62%	–16.49%
2	0.4705	84.13%	73.30%	–10.83%
3	0.3213	87.49%	79.45%	–8.04%
4	0.2440	89.75%	83.37%	–6.38%
5	0.1965	91.35%	86.07%	–5.28%
6	0.1645	92.51%	88.00%	–4.51%
7	0.1414	93.41%	89.48%	–3.93%
8	0.1240	94.10%	90.62%	–3.48%
9	0.1104	94.69%	91.57%	–3.13%
10	0.0995	95.14%	92.30%	–2.84%

4. Approximations of the LRU Hit Ratio

Because the exact LRU result of Equation (12) is tractable only for small caches, the proposed LRU approximations by Fagin [23] and Che et al. [24] are even more relevant. Both approaches are similar: each starts from an equation to determine the mean number of requests $\overline{R}_{LRU}$ until an object O is handed over from the top LRU position to the bottom and leaves the cache if there are no further request to O. $\overline{R}_{LRU}$ also characterizes the cache startup and LRU convergence time until an empty cache of fixed size M is filled [21].

$\overline{R}_{LRU}$ can be determined from a coupon collection process until a series of requests has addressed M different objects. The exact computation of the probability distribution of this process for IRM requests is again scalable only for small caches [21,22,29]. Instead, Che's approximation first determines $\overline{R}_{LRU}$ and then the LRU hit ratio [24,28,37]:

- $\overline{R}_{LRU}$ is approximated by the unique solution of the equation $M = \sum_{j=1}^{N} 1 - e^{-p_j \overline{R}_{LRU}}$.
- Thereafter, the LRU hit ratio is obtained per object (h_{Che} (O_j)) and in total (h_{Che}):

$$h_{Che}(O_j) = 1 - e^{-p_j \overline{R}_{LRU}}; h_{Che} = \sum_{j=1}^{N} p_j h_{Che}(O_j) = \sum_{j=1}^{N} p_j (1 - e^{-p \overline{R}_{LRU}}) \quad (7)$$

Fagin's earlier approach differs from Che's by substituting e^{-pj} with $1 - p$:

$$M = \sum_{k=1}^{N} 1 - (1 - p_k)^{\overline{R}_{LRU}}; h_{\mathrm{Fagin}}\ (O_k) = 1 - (1 - p_k)^{\overline{R}_{LRU}};$$

$$h_{Fagin} = \sum_{k=1}^{N} p_k h_{Fagin}(O_k) = \sum_{k=1}^{N} p_k (1 - (1 - p_k)^{\overline{R}_{LRU}}) \quad (8)$$

Both approximations yield asymptotically exact results for large M, N, as shown in [26–29]. We complement the best-case analysis with a study of the worst-case errors of both approaches [21], similar to Section 3.2 for FIFO. All 204 266 popularity distributions in a 2% raster are checked with $p_k = i_k/50$ for integers i_k ($k \leq N \leq 50$). The approximations h_{Fagin} and h_{Che} are compared with the analysis results via Equation (12) and/or simulations for all cache sizes $M < N$. The study once more strongly suggests the format (6) for distributions with maximum errors of $\Delta h_{\mathrm{Fagin}} = h_{\mathrm{Fagin}} - h_{\mathrm{LRU}}$ and $\Delta h_{\mathrm{Che}} = h_{\mathrm{Che}} - h_{\mathrm{LRU}}$.

Table 3 provides a list of the maximum deviations Δh_{Fagin} and Δh_{Che} for $M \leq 10$, similar to Table 2 for the FIFO deviations. We checked both parameters n, p of Equation (6) for extreme cases and adapted n, p for each M to maximize Δh_{Fagin} and Δh_{Che} [21].

Table 3. Maximum LRU approximation errors of Equations (7) and (8).

Maximum Errors of Che's and Fagin's Approximation for Cache Sizes $M \leq 10$ Worst Case Request Distributions Are of the Type (6) with $p_1 = \cdots = p_n = p/n$						
M	Max. Error $\lvert\Delta h_{\text{Che}}\rvert$	Worst Case (6) n || p/n	h_{IRM}^{LRU} || h_{Che}	Max. Error $\lvert\Delta h_{\text{Fagin}}\rvert$	Worst Case (6) n || p/n	h_{IRM}^{LRU} || h_{Fagin}
1	8.25%	1 || 0.845	0.7055 || 0.6230	Fagin's approximation is exact for $M = 1$		
2	4.48%	2 || 0.455	0.7971 || 0.7523	5.20%	1 || 0.675	0.6041 || 0.6561
3	2.97%	3 || 0.310	0.8247 || 0.7950	3.53%	1 || 0.540	0.4876 || 0.5229
4	2.18%	4 || 0.235	0.8523 || 0.8305	2.82%	2 || 0.360	0.6655 || 0.6937
5	1.71%	5 || 0.192	0.8818 || 0.8647	2.31%	2 || 0.315	0.5867 || 0.6098
6	1.39%	6 || 0.160	0.8922 || 0.8783	1.99%	3 || 0.247	0.6894 || 0.7093
7	1.17%	7 || 0.139	0.9046 || 0.8929	1.72%	3 || 0.227	0.6342 || 0.6514
8	1.03%	6 || 0.155	0.9055 || 0.9158	1.54%	4 || 0.187	0.7033 || 0.7187
9	0.97%	7 || 0.134	0.9188 || 0.9285	1.39%	5 || 0.158	0.7500 || 0.7639
10	0.91%	8 || 0.119	0.9314 || 0.9405	1.26%	5 || 0.150	0.7043 || 0.7169

Note that Fagin's approach [23] is exact for $M = 1$ and that Che et al. [24] report maximum errors of 2% for their approach compared with simulations. We conclude from Table 3:

- The maximum deviations Δh_{Che} of Che's approximation are decreasing with the cache size M from 8.25% for $M = 1$ down to less than 1% for $M \geq 10$.
- The maximum deviations Δh_{Fagin} of Fagin's approximation are decreasing with the cache size M from 5.2% for $M = 2$ down to less than 1.3% for $M \geq 10$.

Figure 3 summarizes the encountered maximum errors for $M = 1, \ldots, 10$:

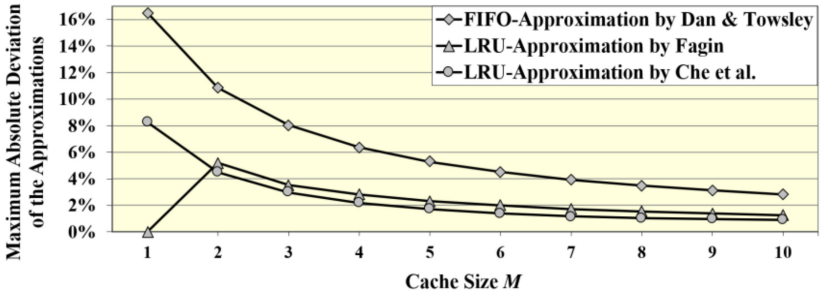

Figure 3. Maximum errors of FIFO and LRU hit ratio approximations.

The LRU results in Table 3 are again based on the evidence from the check of 204 266 popularity distributions in a 2% raster. However, a proof that the distribution type (6) generally maximizes the errors of Fagin's and Che's approaches is open for future study.

We illustrate the evidence in the result $\max(\lvert\Delta h_{\text{Che}}\rvert) \approx 2.97\%$ for $M = 3$. The top-10 error cases out of more than 2×10^5 popularity distributions in the 2% raster are shown in Figure 4. These cases are all in a close surrounding of the extreme case of the type (6), which is shown at the bottom of Figure 4. The top-100 error cases of the list are again in an extended surrounding of the same extreme case. A similar behaviour is observed for all maximum error cases for $M = 1, \ldots, 10$ in Tables 2 and 3.

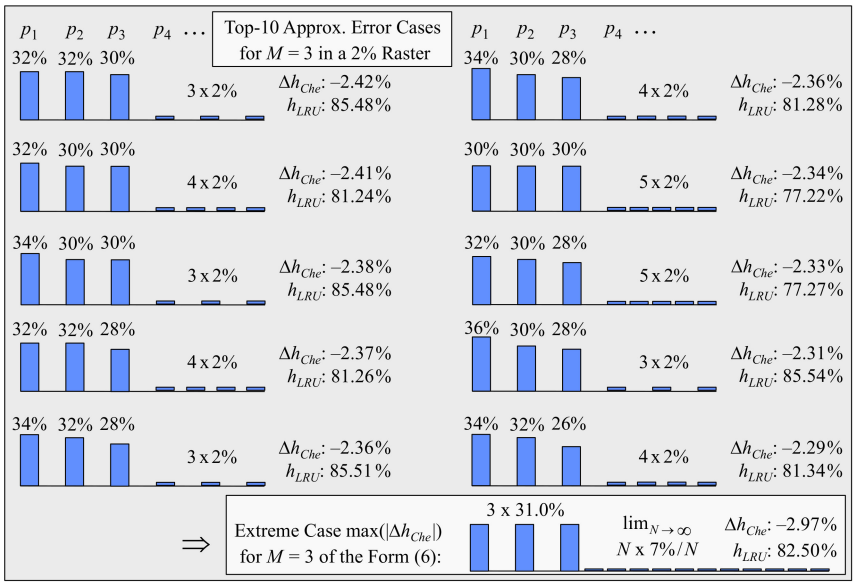

Figure 4. Top-10 error cases for $M = 3$ in a 2% raster and extreme case $\Delta h_{\mathrm{Che}} \approx -2.97\%$.

5. Extended Product Form Solution for Multisegment Caches

The product form solution (1–2) can be extended to caches composed of several levels (lists, partitions, segments) $L_1, \ldots, L_K$. A requested object on cache level L_j ($j \geq 2$) is then forwarded to level L_{j-1} in exchange for an evicted object moving from L_{j-1} to L_j, as illustrated in Figure 5. Cache segmentations can improve the hit ratio by collecting the most popular objects on high cache levels on account of slower adaptation to content changes. Many proposed caching strategies make use of two or more segments, such as ARC [18], segmented LRU [1], k-LRU [38], LRU(m) [12,38], half rank exchange [11], FB-FIFO [39], transposition relocation or CLIMB [7,40]. Distributed architectures also include spreading multilevel storage structures over several cache servers [41].

Garetto et al. [37] and Li et al. [12] include an LRU(K) variant with LRU on the first level L_1. The mixing time is estimated [12] for adaptation to a changing request pattern. An extended product form solution is derived [20] for FIFO(K) and RANDOM(K) strategies when either FIFO or RANDOM is used on all levels. This solution extends to clock per request (CpR) and all variants, which were already included in the single-segment product form of Equations (1) and (2). The solution still remains valid when different strategy variants of the set {FIFO, RANDOM, CpR} are applied on different levels.

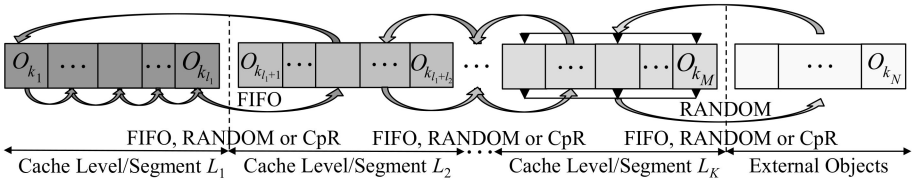

Figure 5. Multisegment caches with FIFO, RANDOM or CpR strategy per segment.

Next, we confirm the generalized product form in an example of two-level caches where FIFO is applied on the first and RANDOM on second level. External objects enter the cache on the second level. Let $p^*(k_1, \ldots, k_m, k_{m+1}, \ldots, k_{m+n})$ denote the steady state IRM probability for content $O_{k_1}, \ldots, O_{k_{m+n}}$ in the cache, where the first cache level ranges from positions $1, \ldots, m$, and the second from $m+1, \ldots, m+n$. Thereafter, we obtain

$$p^*(k_1, \ldots k_{m+n}) = c^*(p_{k_1}, \ldots p_{k_m})^2 (p_{k_{m+1}}, \ldots p_{k_{m+n}}) \tag{9}$$

For a proof, we set up and check the solution (9) in Markovian equilibrium equations. All preceding states are considered on the right, which lead to the cache content

$O_{k_1}, \ldots, O_{k_{m+n}}$ in the next request. A request to an object in the first level keeps the state unchanged according to the FIFO rule. A request to an object on second level moves an object from a position $m+i$ to the top; i.e., $Ok1$ is renewed by such a request with probability $pk1$, as illustrated in Figure 5. In exchange, the object on the bottom of level 1 moves to position $m+i$ on level 2. Upon a cache miss, the requested external object replaces a RANDOM eviction candidate O_{kj} in position $m+i$ on the second level with transition probability $pkm+i/n$. We obtain ($\forall\, j \neq l$: $k_j \neq k_l$)

$$p^*(k_1,\ldots k_{m+n}) = (p_{k_1} + \ldots + p_{k_m})\, p^*(k_1,\ldots k_{m+n})$$

$$+ p_{k_1} \sum_{i=1}^{n} p^*(\overbrace{k_2,\ldots k_m, k_{m+i}}^{Level\ L_1}, \overbrace{k_{m+1},\ldots k_{m+i-1}, k_1, k_{m+i+1},\ldots, k_{m+n}}^{Level\ L_2})$$

$$+ \sum_{i=1}^{n} \frac{p_{k_{m+i}}}{n} \sum_{j=m+n+1}^{N} p^*(\overbrace{k_1,\ldots k_m}^{Level\ L_1}, \overbrace{k_{m+1},\ldots k_{m+i-1}, k_j, k_{m+i+1},\ldots, k_{m+n}}^{Level\ L_2}) \Leftrightarrow$$

$$c^*(p_{k_1},\ldots p_{k_m})^2 (p_{k_{m+1}} \ldots p_{k_{m+n}}) = c^*(p_{k_1} \ldots p_{k_m})^2 (p_{k_{m+1}} \ldots p_{k_{m+n}}) \times$$

$$\Big(\sum_{i=1}^{m} p_{k_i} + \sum_{i=1}^{n} \Big(p_{k_1} \frac{p_{k_{m+i}}}{p_{k_1}} + \sum_{j=m+n+1}^{N} \frac{p_{k_{m+i}}}{n} \frac{p_{k_j}}{p_{k_{m+i}}}\Big)\Big) \Leftrightarrow$$

$$1 = \sum_{i=1}^{m} p_{k_i} + \sum_{i=1}^{n} p_{k_{m+i}} + \sum_{j=m+n+1}^{N} p_{k_j} = \sum_{j=1}^{N} p_{k_j} = 1$$

A check of the ergodicity preconditions is omitted, which would require more space, as well as a proof of the general result: the steady state IRM content distribution for K-level caches with arbitrary combinations of the strategies FIFO, RANDOM and CpR being applied on the levels follows the product form [20] ($\forall\, j \neq l$: $k_j \neq k_l$):

$$\begin{aligned} p^*(k_1,\ldots,k_M) &= c^* \prod_{j=1}^{K} (p_{k_{1+\sum_{i=1}^{j-1} l_i}} \ldots p_{k_{\sum_{i=1}^{j} l_i}})^{K+1-j};\ \sum_{i=1}^{K} l_i = M; \\ h_{IRM}^{K-Level\ Cache} &= \sum_{k_1,\ldots k_M=1}^{N} p^*(k_1,\ldots,k_M) \sum_{j=1}^{M} p_{k_j} \Big/ \sum_{k_1,\ldots k_M=1}^{N} p^*(k_1,\ldots,k_M) \end{aligned} \tag{10}$$

This result includes half rank exchange [11] as a special case with segment sizes $l_j = 2^j$ and $M = 2^{K+1} - 1$. As an extreme case, Climb [7,41] is covered by the multilevel LRU solutions [12,20] and by Equation (10), where $\forall j$: $l_j = 1$. Climb is denoted as transposition relocation in a study by McCabe [7], who compared Climb with LRU in one of the first Markov analysis approaches for caching. Equation (10) still applies when the cache size covers $k \leq K$ levels, such that $M = l_1 + \cdots + l_k$, and the $K - k$ virtual levels are added outside of the cache. The result (10) can be evaluated via a recursive scheme with computational complexity $O(N K^2 (l_1 + 1) \cdots (l_K + 1))$ [20]. Moreover, a mean-field approximation has been derived by Gast and Van Houdt [20] for simpler and faster evaluations.

6. FIFO and LRU Caching Analysis with Data of Different Sizes

The previous results apply to the cache support of CPU/GPU processing in a fixed, unit size data format, whereas web caches are storing files or data chunks of varying sizes. Therefore, we finally regard data objects O_k of different sizes s_k. Then, the cache size M does not refer to the number of objects, but it is measured in bytes, and there is no 1:1 replacement of the data. Often, the caching strategy must perform several evictions until enough space for the new object becomes available. Small objects may be inserted without eviction into free cache space, or several evictions may be required to insert a large object.

To the best of the authors' knowledge, hit ratio solutions for objects of varying sizes are rarely addressed in the literature and are thus open for future study. An extended Che approximation for LRU caches is proposed in Section 3.2 of [28] without further evaluation. We first summarize extended exact solutions for LRU and for the product form in

Sections 6.1 and 6.2. In Sections 6.3–6.5, we extend the FIFO approximation for varying data sizes with regard to unused cache space (UCS) as an additional source of deviations.

6.1. LRU Cache Hit Ratio Solution for Objects of Different Sizes

The IRM steady state probabilities $p_{LRU}(O_{k1}, \ldots, O_{kM})$ of the LRU cache content and the LRU hit ratio h_{IRM}^{LRU} have been derived by King [8] in the following format [12,25]:

$$p_{LRU}(O_k, \ldots O_{k_M}) = \prod_{j=1}^{M} p_{k_j} / \prod_{j=1}^{M-1} (1 - \sum_{i=1}^{j} p_{k_j}) \; (\forall j \neq l : k_j \neq k_l; \forall j : k_j \neq n) \tag{11}$$

$$h_{IRM}^{LRU} = \sum_{k_1,\ldots k_M=1}^{N} p_{LRU}(O_{k_1}, \ldots O_{k_M}) \sum_{j=1}^{M} p_{k_j} = \sum_{n=1}^{N} p_n (p_n + \sum_{m=1}^{M-1} \sum_{k_1,\ldots k_m=1}^{N} p_{LRU}(O_{k_1}, \ldots, O_{k_m}, O_n)) \tag{12}$$

The same result also characterizes cache-filling phases [21]. However, there is no scalable evaluation known for Equations (11) and (12), unlike the computation scheme for the FIFO product form of Section 2.2. Therefore, the solution can be applied only to small caches.

In the format (12), each term of the sum over m refers to hits contributed by the object O_n in LRU stack position $m + 1$. In addition, ${p_n}^2$ covers the hits for O_n in the top position. The solution with differentiation of LRU cache content per stack position is still valid for objects of different sizes, as far as they fit into the cache. This leads to a direct extension of Equation (12) for objects O_n of sizes s_n ($\forall\, j \neq l$: $k_j \neq k_l$; $\forall\, j$: $k_j \neq n$) [21]:

$$\begin{aligned} h_{IRM}^{LRU} &= \sum_{n=1}^{N} p_n (p_n + \sum_{m=1}^{N-1} \sum_{\substack{k_1, \ldots k_m = 1 \\ s_{k_1} + \ldots + s_{k_m} + s_n \leq M}}^{N} p_{LRU}(O_{k_1}, \ldots, O_{k_m}, O_n)) \\ &= \sum_{n=1}^{N} {p_n}^2 (1 + \sum_{m=1}^{N-1} \sum_{\substack{k_1, \ldots k_m = 1 \\ s_{k_1} + \ldots + s_{k_m} + s_n \leq M}}^{N} \prod_{j=1}^{m} \frac{p_{k_j}}{1 - \sum_{i=1}^{j} p_{k_i}}) \end{aligned} \tag{13}$$

6.2. No Common FIFO, RANDOM and CpR Solution for Objects of Different Sizes

In contrast to LRU, the common IRM product form solution of Equation (2) is restricted to a unit data size. Because we did not find an explicit statement on this limitation in the literature, we show that hit ratios of FIFO, RANDOM and CpR can differ already in a simple example for $N = 3$ objects A, B, C with sizes $sA = 1$; $s_B = 2$; and $s_C = 3$ and a cache size of $M = 4$. Then two objects fit together in the cache, except for B and C ($s_B + s_C = 5 > M$).

In this example, the IRM hit ratios of LRU, FIFO, RANDOM and CpR can be analysed by Markov chains, with states for the relevant cache content and state transitions according to the replacement principles of each strategy, as shown in Figure 6. There are six states with one (B or C) or two (A and B, A and C, B and A, C and A) objects as cache content in steady state. Other content states are ignored as transient states (A) or when they exceed the cache size (B and C, A and B and C). The top and bottom cache positions are not distinguished for random evictions, such that four states are sufficient for RANDOM.

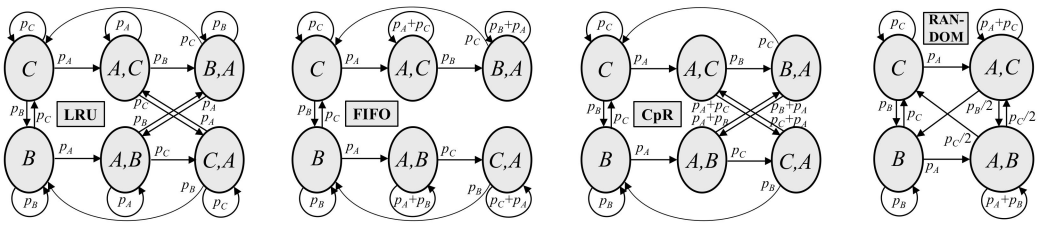

Figure 6. Markov chains for steady state hit ratio analysis of the caching strategies.

The IRM hit ratio analysis first determines the steady state probabilities from equilibrium equations. State-specific hit ratios are given by the sum of the request probabilities of the objects in the cache. The total hit ratio is the sum of state-specific hit ratios weighted by their steady state probabilities. In the case of FIFO, we finally obtain $h_{IRM}^{FIFO} = 1 - 2\ p_B\ p_C/(p_B + p_C) - p_A\ p_B\ p_C/(p_A\ p_B + p_A\ p_C + p_B\ p_C)$. In the case of LRU, the direct computation of the hit ratio via Equation (13) is simpler and equivalent.

The exact analysis shows differences in the hit ratios of the four caching strategies. In the example $p_A = 0.2$, $p_B = 0.7$ and $p_C = 0.1$ we obtain the following:

$$h_{IRM}^{LRU} = \frac{1373}{1800} \approx 76.28\% < h_{IRM}^{FIFO} = \frac{703}{920} \approx 76.41\% < h_{IRM}^{CpR} = \frac{10139}{13240} \approx 76.58\% < h_{IRM}^{RANDOM} = \frac{3109}{4040} \approx 76.96\%$$

The differences in the IRM hit ratio results are not large, but they indicate that several well-known results for unit object size do not hold for data of different sizes:

(1) The FIFO, RANDOM and CpR hit ratios are different for variable data sizes. Their common product form solution (2) [6] is restricted to unit-size objects.
(2) The proven "LRU is better than FIFO under IRM" result [40] is also restricted to unit data size and again violated in the previous example.
(3) Moreover, a monotonous increase in the LRU, FIFO, RANDOM and CpR hit ratio curves (HRC) with the cache size M again holds only for unit data size.

Simulations of large caches with objects of different sizes still show results close to the unit size properties, with negligible differences between FIFO, RANDOM and CpR. However, LRU performance can largely deviate from FIFO, etc. for objects of different sizes. The LRU preference for popular objects leads to a higher hit ratio when those objects are small or to a lower hit ratio when they are large.

Value and Byte Hit Ratio

Steady state Markov results are usually derived for the object hit ratio. We can include a caching value v_k per object for the benefit of serving the object from the cache. v_k may refer to measures for reduced link load and energy consumption on a transport network or to reduced delay, improved throughput and other QoS aspects from the user's perspective. The value hit ratio (VHR) of a cache is defined as ($\forall\ j \neq l$: $k_j \neq k_l$)

$$VHR_{IRM} = \sum_{\substack{k_1,\ldots k_m=1 \\ s_{k_1}+\ldots+s_{k_m}+s_n \leq M}}^{N} p(O_{k_1},\ldots,O_{k_m}) \sum_{j=1}^{m} p_{k_j} v_{k_j} \Big/ \sum_{i=1}^{N} p_i v_i$$

where $p\ (O_{k1}, \ldots, O_{kM})$ again denotes the steady state probability of cache content for a considered strategy. This includes the byte hit ratio (BHR), i.e., the fraction of bytes delivered from the cache when the data size represents the caching value ($v_k = s_k$). In the example of Figure 6, we obtain

$$BHR_{IRM}^{FIFO} \approx 60.61\% < BHR_{IRM}^{LRU} \approx 61.16\% < BHR_{IRM}^{CpR} \approx 61.22\% < BHR_{IRM}^{RANDOM} \approx 61.68\%$$

6.3. Extended Approximations for Objects of Different Sizes

We can straightforwardly extend the FIFO approximation (4) and (5) to data of different sizes s_k:

$$h_{IRM}^{FIFO} = 1 - \sum_{k=1}^{N} 1/(\overline{T}_{InCache} + 1/p_k);\ \overline{T}_{InCache} = M/\sum_{k=1}^{N} s_k/(\overline{T}_{InCache} + 1/p_k) \quad (14)$$

In the fixed size case of Equations (4) and (5), the mean sojourn time $\overline{T}_{InCache}$ of an object in a FIFO cache is determined by a step down of one position per cache miss in the FIFO queue when a new object enters on top. The extension of Equation (14) accounts for a shift down by the size s_k when an object O_k enters the cache. The probability $p_{Enter}(k)$ that O_k is the next object to enter the cache is still assumed to be constant per request:

$$p_{Extern}(k) = p_k \overline{T}_{Extern,k}/(\overline{T}_{InCache} + \overline{T}_{Extern,k}) = 1/(\overline{T}_{InCache} + 1/p_k) \quad (15)$$

where $\overline{T}_{Extern,k}/(\overline{T}_{InCache} + \overline{T}_{Extern,k})$ represents the probability that O_k is outside of the cache. An iterative evaluation of $\overline{T}_{InCache}$ in Equation (14) has similar properties of a monotonous convergence towards a unique solution, as in the unit size case of Equation (5).

The extended approximation of Equation (14) can be used for RANDOM and CpR as well. Their hit ratio slightly differs from FIFO for varying object sizes, where we encounter only minor differences between FIFO, RANDOM and CpR hit ratios, as in the example in Section 6.2, and it is even smaller for large caches.

Che's and Fagin's approximations for the LRU hit ratio are also extensible to objects of different sizes. Again, we have to modify only the iterative equations to determine $\overline{R}_{LRU}$ in Equations (7) and (8) in order to account for object sizes s_k, analogously to the FIFO case of Equation (14) with unchanged formulas for h_{Che} and h_{Fagin} [21,28]:

$$M = \sum_{k=1}^{N} s_k(1 - e^{-p_k\overline{R}_{LRU}});\ h_{Che} = \sum_{k=1}^{N} p_k(1 - e^{-p_k\overline{R}_{LRU}}) \tag{16}$$

$$M = \sum_{k=1}^{N} s_k(1 - (1 - p_k)^{\overline{R}_{LRU}});\ h_{Fagin} = \sum_{k=1}^{N} p_k(1 - (1 - p_k)^{\overline{R}_{LRU}}) \tag{17}$$

6.4. Oversize Objects and Unused Cache Space (UCS)

When the cache is small, there may be oversize objects with $s_k > M$, which cannot enter and therefore should be excluded from the set of relevant data objects in the computations of Equations (14)–(17). A fraction of unused cache space is another factor that is not regarded in those approximations. We estimate its mean value $E[UCS]$ and then apply Equations (14)–(17) with a reduced cache size $M^* = M - E[UCS] = E[F]$, as the mean cache fill level $E[F]$. We transfer an *UCS* estimation scheme for LRU [21] to the FIFO approximation case. The concept of the estimation scheme is summarized in the next section.

6.5. Approximation Scheme for the Mean Unused Space in FIFO Caches

Let F_m denote the cache fill level before the m^{th} request. The fill level stays unchanged for a cache hit: $F_{m+1} = F_m$. In the case of a cache miss, F_m is increasing by the size of the requested object and decreasing by compensating evictions. We obtain

$$F_{m+1} = F_m + s_m^R - \sum_{k=1}^{K} s_{m,k}^E \tag{18}$$

where s_m^R denotes the size of the object of the m^{th} request and $s_{m,k}^E$ are the sizes of eviction candidates. K is the number of required evictions until the requested object fits into the cache, where $K = 0$ if it fits into the currently empty space, i.e., if $F_m + s_m^R \leq M$. The size s_m^R of objects that enter the cache has a discrete distribution $s(j)$ for the number of bytes or kbytes, with a maximum size $s_{\max}$. We obtain $s(j) = \Sigma_k\, p_{Enter}(O_k)\ \mathrm{Prob}(s_k = j)$, where the probability $p_{Enter}(O_k)$ that an object O_k (re-)enters the cache is computed via Equation (15) on the basis of the FIFO approximation. We assign the same discrete distribution $s(j)$ also as the size to the eviction candidates. Finally, the fill level F_m can be evaluated in the format of discrete distributions $f_m(j) = \mathrm{Prob}(F_m = j)$, where $M - s_{\max} \leq F_m \leq M$.

The mean fill level $\mathrm{E}[F]$ is iteratively computed via Equation (18) until steady state is approached when the difference $|\mathrm{E}[F_{m+1}] - \mathrm{E}[F_m]| < \varepsilon$ is becoming negligibly small:

$$\mathrm{E}[F] = \lim_{m \to \infty} E[F_m];\ \mathrm{E}[UCS] = M - \mathrm{E}[F].$$

The entire computation scheme for the mean fill level $\mathrm{E}[F]$ versus unused cache space $\mathrm{E}[UCS]$ is described in detail in [21] for LRU caches. As the main difference in a transfer to FIFO, the probability $p_{Enter}(O_k)$ is obtained via Equation (15), with an impact also on the distribution of the object sizes $s(j)$.

We compare FIFO approximations in the basic format (14) with the previously described scheme for *UCS* correction. Therefore, we consider an example with $N = 50$ objects with a uniform size distribution, i.e., $s_1 = 1, \ldots, s_{50} = 50$, and with independent and Zipf-

distributed requests $p_k = \alpha k^{-\beta}$ for $k = 1, \ldots, 50$, where $\alpha = 1/\Sigma_k\, k^{-\beta}$ [31]. Figure 7 shows the hit ratio curves (HRCs) for three examples:

1. $\beta = 0$, i.e., for uniform requests among all data objects ($p_1 = \ldots = p_{50} = 2\%$),
2. $\beta = 0.8$ with a preference for small objects ($p_1 \approx 15.3\%$, $p_2 \approx 8.8\%$, $\ldots$, $p_{50} \approx 0.67\%$), and
3. $\beta = 0.8$ with a preference for large objects ($p_{50} \approx 15.3\%$, $\ldots$, $p_1 \approx 0.67\%$).

The mean object size per request is $E(S) = 25.5$ for uniform requests, and it is $E(S) \approx 14.1$ or $E(S) \approx 36.9$, respectively, when small or large objects are preferred. A preference for small objects yields the highest hit ratio, whereas uniform requests lead to the lowest hit ratio for cache sizes $M \geq 50$. For $M < 50$, the caching efficiency is reduced because of oversize objects if $s_k = k > M$, with the highest drawback in the third case. The three FIFO HRCs in Figure 7 are obtained via simulation, being accompanied by a curve for the basic FIFO approximation of Equation (14) as FIFO Approx.-1, marked with circles, and the approximation with UCS correction as FIFO Approx.-2, marked with triangles.

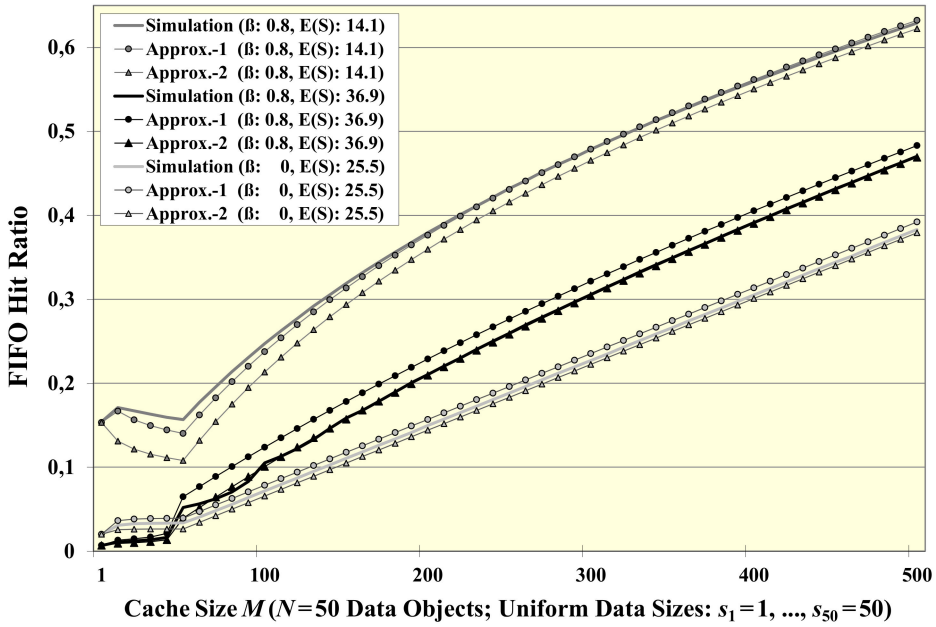

Figure 7. FIFO approximation deviations for a case of uniform data-size distribution.

We conclude that the FIFO approximation can be subject to significant deviations when the cache size M is small in the starting part of the HRC and some popular data objects have a size close to M or larger. The UCS correction can only partly improve the basic FIFO approximation but indicates the extent of corresponding deviations.

We have evaluated the scenario of Figure 7 also for LRU with the basic extended Che approximation [28] and a variant with UCS compensation [21]. The results are very similar to the FIFO results in Figure 7. In the case of $\beta = 0$, the FIFO and LRU HRCs are identical, as are the HRCs for both approximation variants. In the third case with a preference for large objects, LRU and its approximations achieve a slightly higher hit ratio. Up to 6% higher LRU hit ratios are observed for the second case, with the same shape of relative deviations for the approximation variants for LRU as for FIFO.

Although LRU and FIFO caches are frequently applied, it has often been observed [2,17,18] that their web caching performance can be far below the optimum when compared with score- or utility-based methods [10,15]. The latter can use specific information about data items in their caching decisions. Then objects can be preferred in the cache with the highest ratio of popularity to size p_k/s_k as a score to maximize the IRM hit rate according to a knapsack solution.

7. Conclusions

For the exact hit ratio analysis of a cache of fixed size under independent (IRM) requests, the product form (1)–(2) provides the most general solution, which applies to FIFO, RANDOM and clock-based strategies, as well as to combinations of those methods. The product form solution can be computed by the scalable scheme of Equation (3) [30], but it needs an extended number representation, when there are many objects in the cache.

The solution extends to multilevel/multisegment caches [20] with the same set of methods (FIFO, RANDOM, CpR) being applied on each level. However, the hit ratio results differ for those methods if data in a cache is varying in size. The analytic solution 7 for LRU caches even extends to data of different sizes in the form of Equation (13). However, the LRU solutions are not scalable for large caches.

As an alternative, IRM hit ratio approximations have been proposed for LRU by Fagin [23] and Che et al. [24] as well as for FIFO by Dan and Towsley [25]. Those approaches were proven to be asymptotically exact for large caches with data of unit size [26–29]. Moreover, our quantitative analysis identifies the format of Equation (6) for request probabilities with maximum approximation error. The results of the quantitative evaluation lead to error bounds which are summarized as follows:

- The maximum absolute deviations $|\Delta h_{\mathrm{Che}}|$ of Che's approximation are decreasing with the cache size M from 8.25% for $M = 1$ down to less than 1% for $M \geq 10$.
- The maximum absolute deviations $|\Delta h_{\mathrm{Fagin}}|$ of Fagin's approximation are decreasing with the cache size M from 5.2% for $M = 2$ down to less than 1.3% for $M \geq 10$.
- The maximum deviations of the FIFO approximation [25] are decreasing with the cache size M from 16.5% for $M = 1$ down to less than 3% for $M \geq 10$.

Extensions of those approximations for data of different sizes are provided [21,28], which are still expected to yield asymptotic exact results for large caches. However, a fraction of unused cache space contributes to larger errors for small caches. Therefore, an exact analysis is recommended to avoid the largest approximation errors for small caches, as far as Markovian and other exact solutions are scalable.

The presented framework of analysis and approximation results provides a basic tool set for a performance evaluation of content delivery in current CDNs and cloud architectures, as well as in ICN, CCN and NDN concepts towards a future Internet. The limited scope of exact solutions and insufficient accuracy of approximations for varying data sizes are major challenges for future research.

Author Contributions: Formal analysis, G.H., K.N. and O.H.; software, F.H. and K.N.; methodology, G.H. and O.H.; writing—first draft, G.H.; writing—review and editing, all authors. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable, the study does not report any data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Podlipnik, S.; Böszörmenyi, L. A survey of web cache replacement strategies. *ACM Comput. Surv.* **2003**, *35*, 374–398. [CrossRef]
2. Arlitt, M.; Williamson, C. Internet web servers: Workload characterization and performance implications. *IEEE Trans. Netw.* **1997**, *5*, 631–645. [CrossRef]
3. Li, S.; Xu, J.; van der Schaar, M.; Li, W. Popularity-driven content caching. In Proceedings of the IEEE Infocom, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9.
4. Paschos, G.S.; Iosifidis, G.; Tao, M.; Towsley, D.; Caire, G. The role of caching in future communication systems and networks. *IEEE JSAC* **2018**, *36*, 1111–1125. [CrossRef]
5. Corbato, F.J. A paging experiment with the Multics system. *MIT Proj. MAC Rep. MAC-M-384* **1968**, 1–14.

6. Gelenbe, E. A unified approach to the evaluation of a class of replacement algorithms. *IEEE Trans. on Comp.* **1973**, *100*, 611–618. [CrossRef]
7. McCabe, J. On serial files with relocatable records. *Oper. Res.* **1965**, *13*, 609–618. [CrossRef]
8. King, W.F., III. Analysis of demand paging algorithms. In Proceedings of the IFIP Congress, Ljubljana, Yugoslavia, 23–28 August 1971; pp. 485–490.
9. 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.* **2001**, *50*, 1352–1361.
10. Dehghan, M.; Massoulie, L.; Towsley, D.; Menasche, D.S.; Tay, Y.C. A Utility optimization approach to network cache design. *IEEE/ACM Trans. Netw.* **2019**, *27*, 1013–1027. [CrossRef]
11. Hasslinger, G.; Ntougias, K.; Hasslinger, F.; Hohlfeld, O. Performance evaluation for new web caching strategies combining LRU with score-based selection. *Comput. Netw.* **2017**, *125*, 172–186. [CrossRef]
12. Li, J.; Shakkottai, S.; Lui, J.C.S.; Subramanian, V. Accurate learning or fast mixing? Dynamic adaptability of caching algorithms. *IEEE JSAC* **2018**, *36*, 1314–1330. [CrossRef]
13. Paschos, G.S.; Iosifidis, G.; Caire, G. Cache optimization models and algorithms. *Found. Trends Commun. Inf. Theory* **2020**, *16*, 156–345. [CrossRef]
14. Eytan, O.; Harnik, D.; Ofer, E.; Friedman, R.; Kat, R. It's time to revisit LRU vs. FIFO. In Proceedings of the 12th USENIX HotStorage Workshop, Berkeley, CA, USA, 13–14 July 2020; pp. 1–7.
15. Hasslinger, G.; Ntougias, K.; Hasslinger, F.; Hohlfeld, O. Fast and efficient web caching methods regarding the properties per data. In Proceedings of the IEEE CAMAD, Limassol, Cyprus, 11–13 September 2019; pp. 1–7.
16. Yang, J.; Yue, Y.; Rashmi, K.V. A large-scale analysis of key-value cache clusters at Twitter. *ACM Trans. Storage* **2021**, *17*, 1–35. [CrossRef]
17. ElAarag, H. *Web Proxy Cache Strategies: Simulation, Implementation and Performance Evaluation*; Springer Publisher: Berlin/Heidelberg, Germany, 2013; pp. 1–103.
18. Megiddo, N.; Modha, S. Outperforming LRU with an adaptive replacement cache algorithm. *IEEE Comput.* **2004**, *37*, 58–65. [CrossRef]
19. Ntougias, K.; Papadias, C.B.; Papageorgiou, G.K.; Hasslinger, G.; Sorensen, T.B. Coordinated caching and QoS-aware resource allocation for spectrum sharing. *Wirel. Pers. Comm.* **2020**, *112*, 49–79. [CrossRef]
20. Gast, N.; Van Houdt, B. Transient and steady-state regime of a family of list-based cache replacement algorithms. In Proceedings of the ACM Sigmetrics, Portland, OR, USA, 15–19 June 2015.
21. Hasslinger, G.; Ntougias, K.; Hasslinger, F.; Hohlfeld, O. Analysis of the LRU cache startup phase and convergence time and error bounds on approximations by Fagin and Che. In Proceedings of the WiOpt Symposium, CCDWN workshop, Turin, Italy, 19 September 2022; pp. 1–8.
22. Wong, K.Y.; Yeung, A.; Choi, K.C.; Lei, P.; Lam, C.T. Exact transient analysis on LRU cache startup for IoT. In Proceedings of the 9th International Conference on Information Technology: IoT and Smart City, New York, NY, USA, 22–25 December 2021; pp. 310–315.
23. Fagin, R. Asymptotic miss ratios over independent references. *J. Comp. Syst. Sci.* **1977**, *14*, 222–250. [CrossRef]
24. Che, H.; Tung, Y.; Wang, Z. Hierarchical web caching systems: Modeling, and experimental design. *IEEE JSAC* **2002**, *20*, 1305–1314.
25. Dan, A.; Towsley, D. An approximate analysis of the LRU and FIFO buffer replacement schemes. In Proceedings of the ACM SIGMETRICS, Boulder, CO, USA, 22–25 May 1990; pp. 143–152.
26. Berthet, C. Approximation of LRU caches miss rate: Application to power-law popularities. *arXiv* **2017**, arXiv:1705.10738.
27. Brenner, M. A Lyapunov analysis of LRU. Master Thesis, University of Illinois Urbana-Champaign, Champaign, IL, USA, 2020; pp. 1–42.
28. Fricker, C.; Robert, P.; Roberts, J. A versatile, accurate approximation for LRU cache performance. In Proceedings of the ITC 24, Krakow, Poland, 4–7 September 2012; pp. 1–8.
29. Poojary, P.; Moharir, S.; Jagannathan, K. A coupon collector based approximation for LRU cache hits for Zipf requests. In Proceedings of the 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt), Philadelphia, PA, USA, 18–21 October 2021; pp. 1–8.
30. Fagin, R.; Price, T.G. Efficient calculation of expected miss ratios in the independent reference model. *SIAM J. Comput.* **1978**, *7*, 288–296. [CrossRef]
31. Breslau, L.; Cao, P.; Fan, L.; Phillips, G.; Shenker, S. Web caching and Zipf-like distributions: Evidence and implications. In Proceedings of the IEEE Infocom, New York, NY, USA, 21–25 March 1999; pp. 126–134.
32. Traverso, S.; Ahmed, M.; Garetto, M.; Giaccone, P.; Leonardi, E.; Niccolini, S. Unraveling the impact of temporal and geographical locality in caching systems. *IEEE Trans.* **2015**, *17*, 1839–1854.
33. Hasslinger, G.; Kunbaz, M.; Hasslinger, F.; Bauschert, T. Web caching evaluation for Wikipedia request statistics. In Proceedings of the IEEE WiOpt Symposium, Paris, France, 15–19 May 2017; pp. 1–6.
34. Rosensweig, E.J.; Menasche, D.S.; Kruose, J. On the steady-state of cache networks. In Proceedings of the IEEE Infocom, Turin, Italy, 14–19 April 2013; pp. 863–871.

35. Marin, A.; Rossi, S.; Burato, D.; Sina, A.; Sottana, M. A product-form model for the performance evaluation of a bandwidth allocation strategy in WSN. *ACM TOMACS* **2018**, *28*, 1–23. [CrossRef]
36. Tsukada, N.; Hirade, R.; Miyoshi, N. Fluid limit analysis of FIFO and RR caching for IRM. *Perform. Eval.* **2012**, *69*, 403–412. [CrossRef]
37. Garetto, M.; Leonardi, E.; Martina, V. A unified approach to the performance analysis of caching systems. *ACM Trans. Model. Perform. Eval. Comput. Syst.* **2016**, *1*, 1–28. [CrossRef]
38. Gast, N.; Van Houdt, B. TTL approximations of the cache replacement algorithms LRU(m) and h-LRU. In *Performance Evaluation*; Elsevier: Amsterdam, The Netherlands, 2017; pp. 33–57.
39. Gomaa, H.; Messier, G.G.; Williamson, C.; Davies, R. Estimating instantaneous cache hit ratio using Markov chain analysis. *IEEE/ACM Trans. Netw.* **2013**, *21*, 1472–1483. [CrossRef]
40. Van den Berg, J.; Gandolf, A. LRU is better than FIFO under the independent reference model. *J. Appl. Probab.* **1992**, *29*, 239–243. [CrossRef]
41. Starobinsky, D.; Tse, D. Probabilistic methods for web caching. *Perf. Eval.* **2001**, *46*, 125–137. [CrossRef]