Markov Analysis for Optimum Caching as an Alternative to Belady's Algorithm

Gerhard Hasslinger, Deutsche Telekom, Darmstadt, Germany

✉ gerhard.hasslinger@telekom.de

- Analytic Results on LRU, LFU, Optimum Caching
- Belady's Principle for Optimum Caching [L.A. Belady 66]
- Markov Analysis: Cache of size $M = 1$ for a single object
- Comparison to known results [A. Smith 76], [D.E. Knuth 85]
- General Markov Analysis compared to Belady's Algorithm

Hit Rate for Least Recently Used (LRU)

- Hit rate analysis for N objects, cache size M, assuming independent requests (IRM) with prob. $p_1, \ldots, p_N$ per object
- Exact hit rate result, see e.g. [Dan & Towsley 1999]:

$$h_{M,N}^{\mathrm{LRU}} = \sum_{\substack{k_1,\ldots,k_M=1; \\ i\neq j \Rightarrow k_i \neq k_j}}^{N} \Pr\{o_{k_1},\ldots,o_{k_M} \in C\} \sum_{j=1}^{M} p_{k_j}$$

$$= \sum_{k_1=1}^{N} p_{k_1} \sum_{\substack{k_2=1 \\ k_2\neq k_1}}^{N} \frac{p_{k_2}}{1-p_{k_1}} \sum_{\substack{k_3=1 \\ k_3\neq k_1,k_2}}^{N} \frac{p_{k_3}}{1-p_{k_1}-p_{k_2}} \cdots \sum_{\substack{k_M=1 \\ k_M\neq k_1,\ldots,k_{n-1}}}^{N} \frac{p_{k_M}}{1-\sum_{j=1}^{M-1} p_{k_j}} \sum_{j=1}^{M} p_{k_j}$$

- Complexity: $O\binom{N}{M}$ (State cover all objects in the cache)
- Che approximation [Che02] or simulation is used for LRU hit rate evaluation for moderate and large N, M.

Hit rate for Least Frequently Used (LFU)

- Hit rate analysis for N objects, cache size M, assuming independent requests (IRM) with probabilities $p_1 \geq \ldots \geq p_N$
- The general result is:

$$h_{M,N}^{\mathrm{LFU}} = \sum_{k=1}^{M} p_k$$

- assuming M objects with maximum request probabilities being constantly kept in the cache
- Maximum IRM hit rate without prediction or look-ahead
- LFU statistics over few hours or a day is most relevant

Belady's Algorithm for Optimum Caching

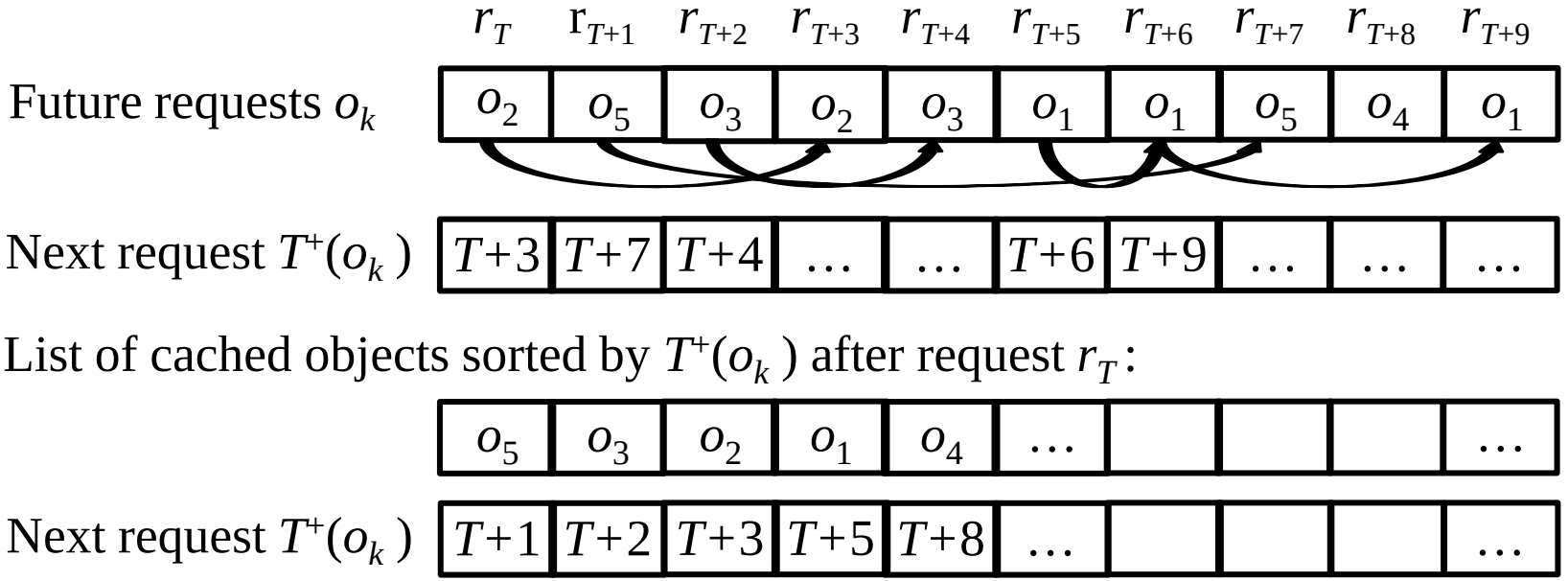
Based on knowledge of future requests:

Replace the object with farthest next request ↔

Keep/insert objects with the next request (i.e. hit) in a sorted list

Effort per request for sorting: $O(\ln(M))$

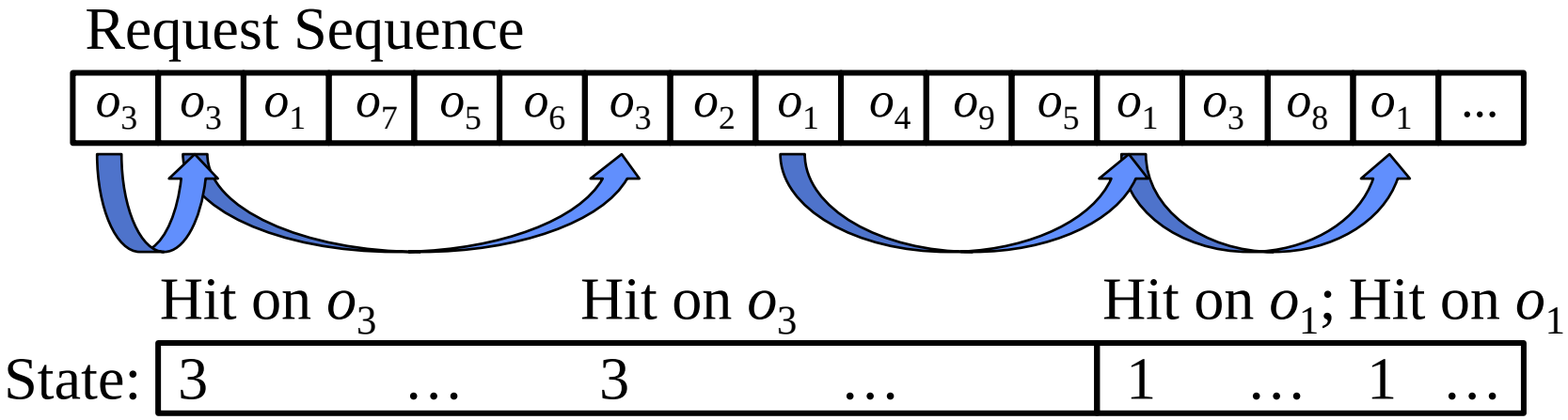
Precompute sequence of "next request" per request



Analysis of Optimum Caching: Cache Size $M = 1$

Markov analysis for $M = 1$:

- State indicates the object with the lastest hit
- State k changes, if another object is requested twice before o_k
- Inter hit time: Request sequence to pairwise different objects
- State transition matrix via combinatorial formulas



Analysis of Optimum Caching: Cache Size $M = 1$

- A Markov model with N states is sufficient for N objects and for independent requests with prob. $p_1, \ldots, p_N$ per object
- The state marks the object o_k addressed by the last cache hit

$$\pi_m(O_n) = \Pr\{m \text{ requests to pairwise different objects from the set } O_n\}$$

$$= m! \sum_{\substack{k_1 =,\ldots,k_m \le n \\ j \ne l \Rightarrow k_j \ne k_l}} p_{k_1} p_{k_2} \ldots p_{k_m}; \quad \pi_m(O_n) = \pi_m(O_{n-1}) + \pi_{m-1}(O_{n-1})\, m\, p_n$$

- $q_{kk}^m = p_k \pi_m(O/\{o_k\}) = \Pr\{\text{Next hit on } o_k \text{ after } m+1 \text{ requests}\}$
- $q_{kj}^m = (m+1) p_j^2 \pi_m(O/\{o_k, o_j\}) = \Pr\{\text{Next hit on } o_j \text{ after } m+2 \text{ req.}\}$
- **Transition matrix of the Markov chain**:

$$q_{kk} = p_k \sum_{m=0}^{N-1} \pi_m(O/\{o_k\}); \quad q_{kj} = \sum_{m=0}^{N-2} q_{kj}^m \quad \text{for } k \ne j.$$

- Steady state solution $r_k = \Pr\{\text{A hit is addressing } o_k\} = \sum_{j=1}^{N} r_j q_{jk}$
- Hit rate via mean **inter hit time**: $h_{\text{Opt}} = 1 \Big/ \sum_{k=1}^{N} r_k \left(1 + \sum_{m=1}^{N-1} \pi_m(O/\{o_k\})\right)$

Analysis of Optimum Caching: Cache Size $M = 1$

- The previous solution of the N state Markov chain has complexity $O(N^4)$ for computing all transition probabilities
- Fortunately, we can prove a **simple steady state solution**:

$$r_j = \frac{p_j^2}{\sum_{k=1}^{N} p_k^2}; \qquad h_{\mathrm{OPT}} = \frac{\sum_{k=1}^{N} p_k^2}{\sum_{k=1}^{N} p_k^2 E(R_k)} = \frac{\sum_{k=1}^{N} p_k^2}{\sum_{k=1}^{N} p_k^2 \left(1+\sum_{m=1}^{N-1} \pi_m(O/\{o_k\})\right)}$$

  i.e. the probability of a hit on o_k is proportional to p_k^2

- This reduces the computational complexity to $O(N^2)$

Result for Paging Systems with Uniform Requests & $M = 2$

An Analysis of Optimum Caching

DONALD E. KNUTH*

$$\cdots$$

$$Q(n) = 1 + \frac{n-1}{n} + \frac{n-1}{n}\frac{n-2}{n} + \cdots,$$

is Ramanujan's function studied in [5, Section 1.2.11.3]. For general d, a similar derivation gives $x_1 = 1/Q(d-1)$, so we have

$$f(2,d) = \frac{d-1}{d}\left(1 - \frac{1}{Q(d-1)}\right).$$

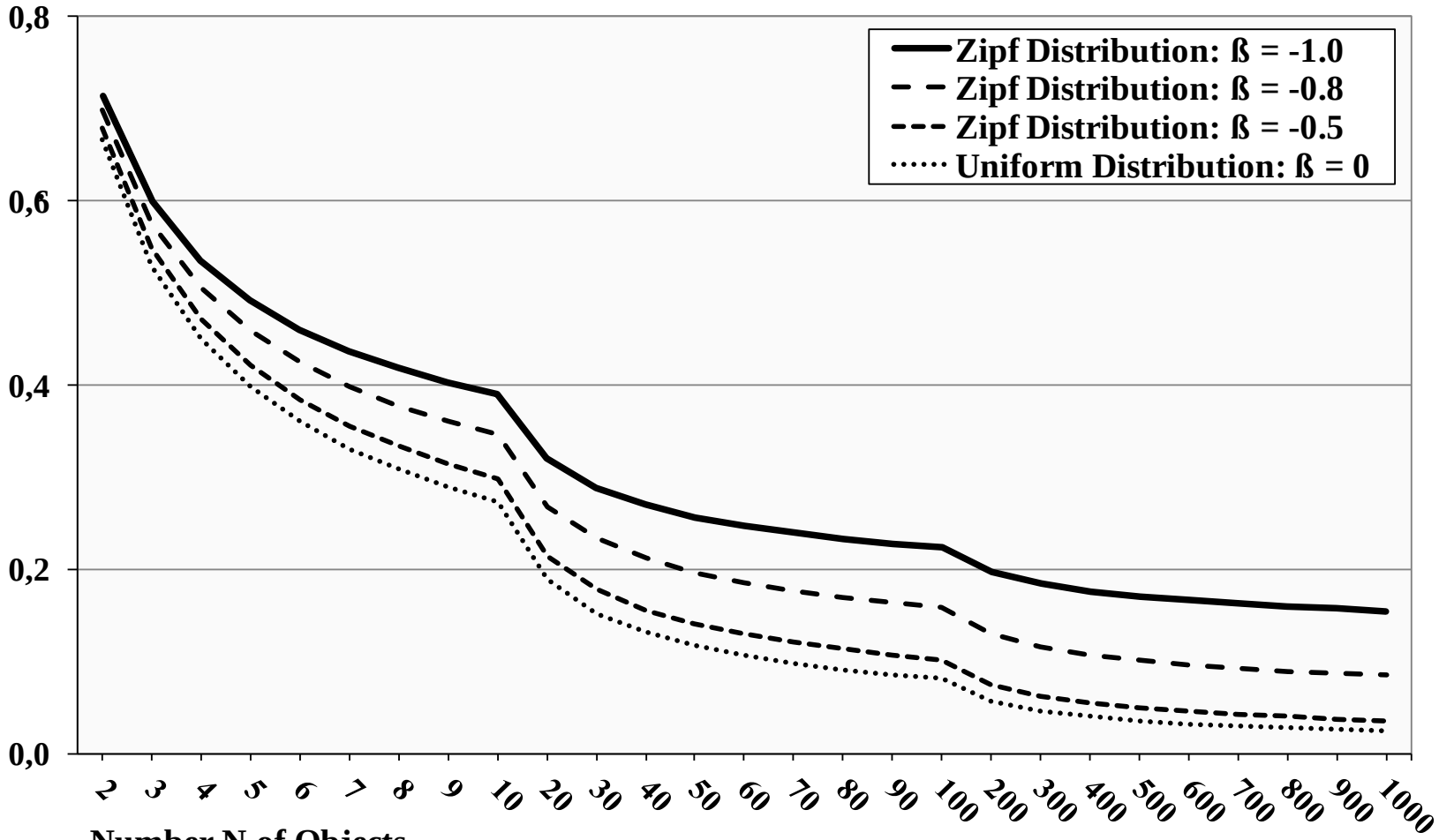
This formula is surprising on two counts. In the first place, the probability $1 - f(2,d)$ of nonfailure is $(d-1)/(dQ(d-1)) + 1/d$, which is $\sqrt{2/(\pi d)} + O(1/d)$ by [5, Eq. 1.2.11.3–25]. This is roughly $\sqrt{d}$ times the nonfailure probability obtained without a clairvoyant cache, even though our cache has only two entries, so the small cache is doing unexpectedly well

Conversion: Paging systems ↔ Web cache results:

$$h_{\text{PagingCache}}(N+1, M+1) = [(1 + N\, h_{\text{WebCache}}(N, M)] / (N+1)$$

$$h_{\text{Opt, WebCache}} = 1/\sum_{k=0}^{N-2} N^{-k}(N-1)!/(N-k-1)!$$

Analysis of Optimum Caching: Cache Size $M = 1$



Example of the General Markov Model for Optimum Caching Analysis ($M = 3$)

Markov model states include
all objects which can lead to a hit in the next request
(**Cached objects** + loading options)

Request Sequence		Initial cache content → **1** **2** **3**
4	(*Miss*)	{4} **1** **2** **3**
5	(*Miss*)	{5, 4} **1** **2** **3**
2	(Hit_1)	**2** {5, 4} **1** **3**
6	(*Miss*)	{6} **2** {5, 4} **1** **3**
1	(Hit_1)	**1** {6} **2** {5, 4} **3**
7	(*Miss*)	{7} **1** {6} **2** {5, 4} **3**
4	(Hit_4)	**4** {7} **1** {6} **2**
5	(*Miss*)	{5} **4** {7} **1** {6} **2**
5	(Hit_3)	**5** {4, 7} **1** {6} **2**
7	(Hit_3)	**7** **5** {4, 1, 6} **2**
3	(*Miss*)	{3} **7** **5** {4, 1, 6} **2**
6	(Hit_4)	**6** {3} **7** **5**
7	(Hit_1)	**7** **6** {3} **5**
5	(Hit_2)	**5** **7** **6**
…	…	…

Indices of the requested objects

Markov Analysis ↔ Belady´s Algorithm

Object requested at r_T and (transition type)			State S_T represented as $L_1, ok_1, L_2, ok_2, \ldots, L_M, ok_M$	Cache content for optimum caching at r_T
Initial cache content →			**1 2 3**	**1 2 3**
$T = 1$	4	(*Miss*)	{4} **1 2 3**	**1 2 4**
↓ 2	5	(*Miss*)	{5, 4} **1 2 3**	**1 2 4**
3	2	(Hit_1)	**2** {5, 4} **1 3**	**1 2 4**
4	6	(*Miss*)	{6} **2** {5, 4} **1 3**	**1 6 4**
5	1	(Hit_1)	**1** {6} **2** {5, 4} **3**	**1 6 4**
6	7	(*Miss*)	{7} **1** {6} **2** {5, 4} **3**	**7 6 4**
7	4	(Hit_4)	**4** {7} **1** {6} **2**	**7 6 4**
8	5	(*Miss*)	{5} **4** {7} **1** {6} **2**	**7 6 5**
9	5	(Hit_3)	**5** {4, 7} **1** {6} **2**	**7 6 5**
10	7	(Hit_3)	**7 5** {4, 1, 6} **2**	**7 6 5**
11	3	(*Miss*)	{3} **7 5** {4, 1, 6} **2**	**7 6 5**
12	6	(Hit_4)	**6** {3} **7 5**	**7 6 5**
13	7	(Hit_1)	**7 6** {3} **5**	**7 6 5**
14	5	(Hit_2)	**5 7 6**	**7 6 5**
…	…	…	…	…

Conclusions on Markov Analysis for Optimum Caching

Size of the Markov model is large:

$$\left|S_T\right| = N(N-1)\cdots(N-M+1)\cdot(M+1)^{N-M}$$

Comparable analysis model by A. Smith [Smi76]:

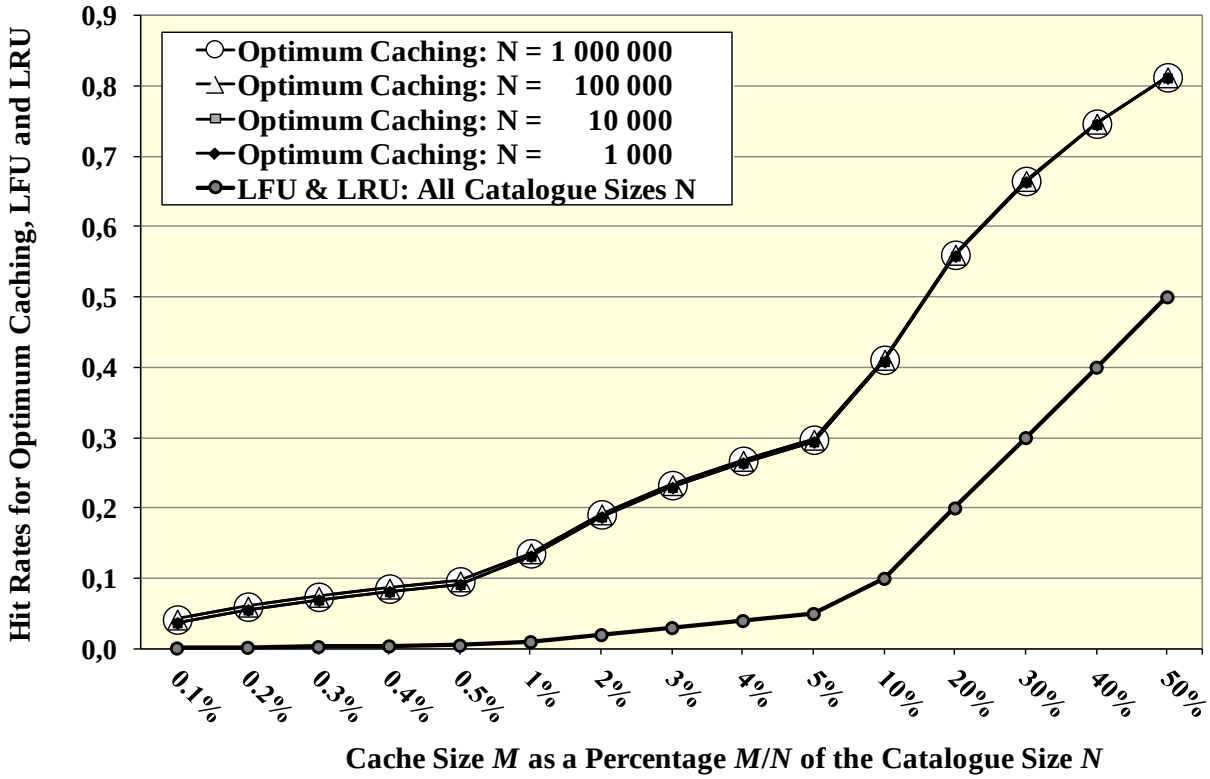
$$\left|S_{Smith}\right| = N!\binom{N}{M} = \frac{N(N-1)\cdots(M+1)}{(M+1)^{N-M}}\left|S_T\right|$$

Direct analysis approaches are limited to $N \leq 20$

Markov Analysis is an alternative to Belady's algorithm

- Effort per request is similar for both approaches
- Markov analysis decides about hits per request without look-ahead (!)
  $\leftrightarrow$ L.A. Belady: "The optimal solution can be found by storing the program's entire sequence of references and then working backward to reconstruct a minimum replacement sequence."
- Limited look-ahead is relevant for caches in large CDNs

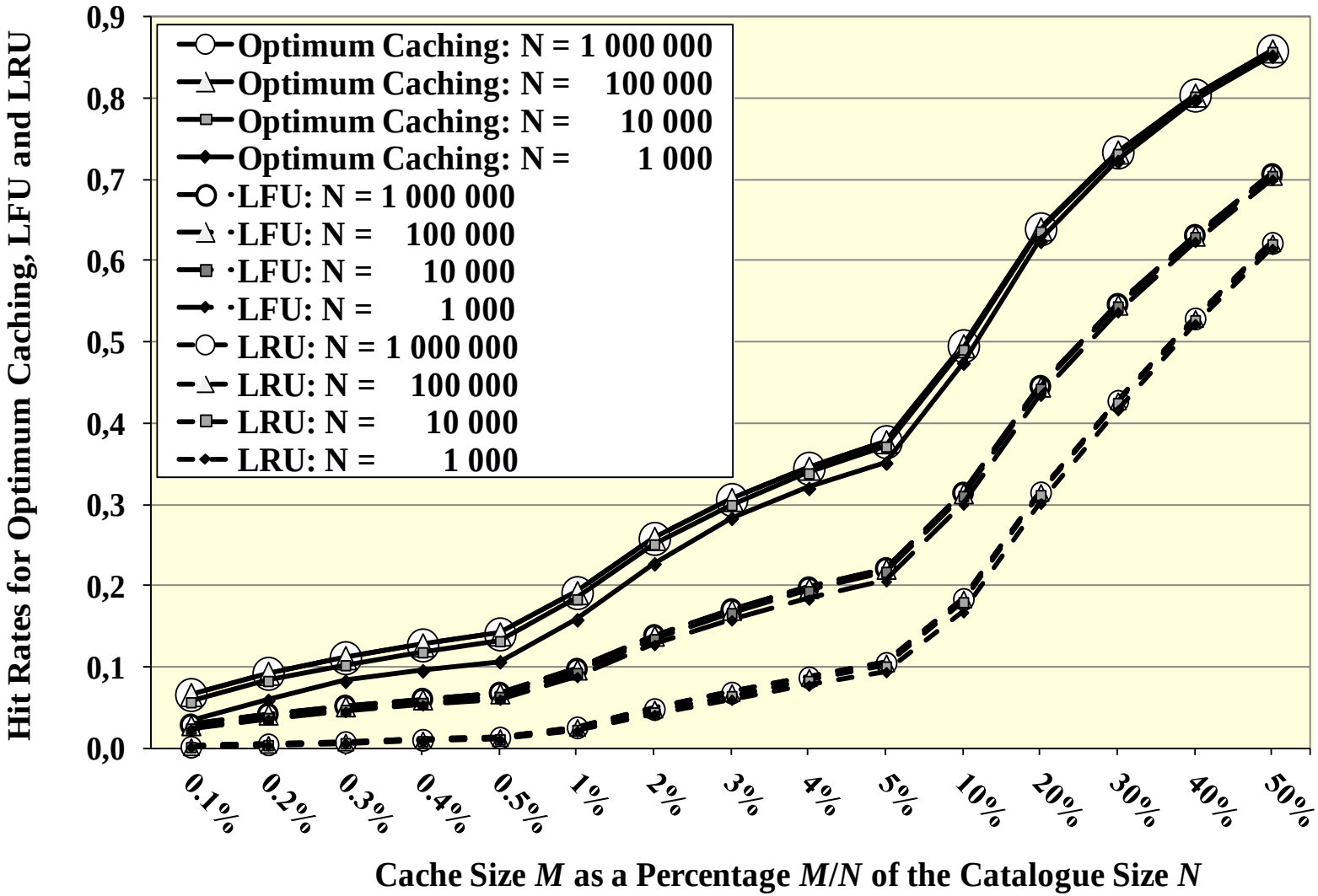
Results for independent uniform requests



Uniform requests:
Request
Probabilities
$1/N$
for all N objects

- For optimum caching we obtain $1.37\sqrt{M/N} > h_{Opt} > 1.15\sqrt{M/N}$
- For LRU, SG-LRU, LFU we obtain $h = M/N$ as worst case scenario

Results for Zipf law requests $Z(k) = \alpha\, k^{\beta}$ with $\beta = -0.5$



References

[Bel66] L.A. Belady, A study of replacement algorithms for a virtual-storage computer, IBM Systems Journal 2 (1966) 78-101

[BS17] N. Beckmann and D. Sanchez, Maximizing cache performance under uncertainty, Proc. 23rd Symposium on High Perform. Comp. Arch. (2017)

[Che02] H. Che, Y. Tung, and Z. Wang, Hierarchic web caching systems: modeling, design and experimental results, IEEE JSAC 20(7) (2002) 1305-14

[Ha17a] G. Hasslinger et al., Performance evaluation for new web caching strategies combining LRU with score based selection, Computer Networks (2017)

[Ha17b] G. Hasslinger et al., Web Caching Evaluation from Wikipedia Request Statistics, Symposium on Optimization in Wireless Networks, Paris (2017)

[Ha17c] G. Hasslinger, Markov Analysis for Optimum Caching as an Alternative to Belady's Algorithm, submitted to MMB'18 <www.mmb2018.de>

[Knu85] D.E. Knuth, An analysis of optimum caching, Journal of algorithms (1985)

[JL16] A. Jain and C. Lin, Back to the future: Leveraging Belady's algorithm for improved cache replacement, in Proc. ISCA-4,(2016)

[Smi76] A.J. Smith, Analysis of the optimal, look-ahead demand paging algorithms, SIAM Journal Computation 5/4 (1976) 743-757