Hit Rate vs. Hit Probability Based Cache Utility Maximization*

Nitish K. Panigrahy†, Jian Li† and Don Towsley
University of Massachusetts Amherst
{nitish, jianli, towsley}@cs.umass.edu
†Co-primary authors

1. INTRODUCTION

Caches play a prominent role in networks and distributed systems, and their importance is reflected in much recent work on performance analysis of caching algorithms. A plethora of research has been done on the analysis of caching algorithms using the metric of hit probability under the Independent Reference Model (IRM). However, there has been a tremendous increase in the demand of different types of content with different quality of service requirements; consequently, the user needs in the networks become more heterogeneous. In order to meet such challenges, the design and analysis of content delivery networks need to incorporate service differentiation among different classes of contents and applications. Though considerable literature has focused on the design of fair and efficient caching algorithms for content distribution, little work has focused on the provision of multiple levels of service in network and web caches.

A related problem has been considered in [1], where the authors formulated a *Hit-probability Based Cache Utility Maximization* (HPB-CUM) framework under the IRM and constant content size

$$\textbf{HPB-CUM}: \quad \max \sum_{i=1}^{n} U_i(h_i^p), \quad \text{s.t.} \sum_{i=1}^{n} h_i^p = B, \qquad (1)$$

where n is the number of unique contents in the system, B is the cache size, h_i^p is the stationary hit probability of content i, and $U_i : [0,1] \to \mathbb{R}$ is the utility function. The paper characterized the optimal TTL cache policies [3] with an increasing, continuously differentiable, and strictly concave utility function, and also proposed an online algorithm for cache management.

While the characterization of cache management with respect to (w.r.t.) hit probability is valuable, *hit rate* [2] is a more generic performance metric given the request arrival rate in real systems. For example, pricing based on request rate is preferable to that based on cache occupancy by a service provider. Furthermore, the goal of a service provider in designing hierarchical caches might be to minimize the internal bandwidth cost, which can be characterized with a utility function $U_i = -C_i(m_i)$, where $C_i(m_i)$ is the cost associated with miss rate m_i for content i. Hence, it is insufficient to only identify the optimal cache management w.r.t. hit probability, and one also needs to characterize the optimal policy when utility is measured as a function of hit rate.

Therefore, it is reasonable to define utility functions as functions of hit rates. Then a fundamental research question is: *when hit rate based utility maximization is favorable than hit probability based utility maximization?* One argument in support of utility maximization as functions of hit rate is that it is more natural from the perspective of content providers. Since the utility function should not only capture the impact of hit probability, but also the amount of data arriving at the system, which is characterized by the hit rate. However, performance analysis results on hit rate with utility maximization are relatively few. The typical performance analysis under the IRM usually follows a fixed Zipf distribution that models a heavy tail popularity distribution observed in empirical studies. One objective in this paper is to formulate a *Hit-rate Based Cache Utility Maximization* (HRB-CUM) framework for maximizing aggregate content utility subject to buffer size constraints at the service provider. We wish to explore the tradeoff between HRB-CUM and HPB-CUM: (i) adaptability to the heavy-tailed request process, and (ii) the robustness and stability of the corresponding online-algorithms.

Towards this goal, we first formulate HRB-CUM and then derive explicit expressions for the corresponding optimal hit rate and hit probability under the family of β-fair utility functions. We compare the relative metrics under different content weights. We find that under IRM requests, there exists a threshold on the identity of content. To be more specific, if we assume that the popularity is in a non-increasing order, i.e., $p_1 \geq \cdots \geq p_n$, then for $\beta < 1$, there exists a threshold $1 < j < n$ such that HRB-CUM will favor most popular contents than HPB-CUM, i.e., popular contents will be cached under HRB-CUM. Similar arguments for $\beta > 1$, and these will be described in details in Section 3.

Although the above comparisons provide insights on the advantage of HRB-CUM over HPB-CUM in the notion of fairness, they say nothing about how well one approach can respond to the changes in the system. The relevant methodology is to propose online algorithms that can adapt to changes in the system with limited information. We show that the corresponding online algorithms for HRB-CUM is more robust and stable than HPB-CUM w.r.t. the convergence rate, details are given in Section 4.

2. MATHEMATICAL MODELS

We consider the IRM model with a Poisson arrival pro-

*This research was sponsored by the U.S. ARL and the U.K. MoD under Agreement Number W911NF-16-3-0001 and by the NSF under grant NSF CNS-1617437.

cess for most of our analysis. Suppose there are a set of n contents and a cache of size B. Let $\lambda_i^r = \lambda_i h_i^r$ denote the hit rate for content i with arrival rate λ_i and hit probability h_i^r.

TTL Caches: Under the TTL cache policy, content i is inserted into the cache with a timer t_i at the time of a cache miss. In particular, we consider the reset TTL cache, i.e., the TTL is reset to t_i each time content i is requested. From previous work [2], the hit rate of content i satisfies $\lambda_i^r = \lambda_i(1 - e^{-\lambda_i t_i})$.

Utility Function and Fairness: Different utility functions define different fairness properties. Here, we focus on the widely used β-fair utility functions, defined as follows

$$U_i(x) = \begin{cases} w_i \frac{x^{1-\beta}}{1-\beta}, & \beta \geq 0, \beta \neq 1; \\ w_i \log x, & \beta = 1, \end{cases} \quad (2)$$

where $w_i > 0$ denotes the weight for content i.

Cache Utility Maximization: We formulate cache management as a utility maximization problem. We introduce HRB-CUM in this section and compare it to HPB-CUM given in (1) [1]. More specifically,

$$\textbf{HRB-CUM}: \quad \max \sum_{i=1}^{n} U_i(\lambda_i^r), \quad \text{s.t.} \sum_{i=1}^{n} \lambda_i^r/\lambda_i = B. \quad (3)$$

With the Lagrangian method, we easily obtain the optimal hit rate λ_i^r and hit probability h_i^r under HRB-CUM for $\beta \geq 0$ and $\beta \neq 1$ given as follows

$$\lambda_i^r = \frac{w_i^{1/\beta}\lambda_i^{1/\beta}}{\sum_j w_j^{1/\beta}\lambda_j^{1/\beta-1}}B, \quad h_i^r = \frac{w_i^{1/\beta}\lambda_i^{1/\beta-1}}{\sum_j w_j^{1/\beta}\lambda_j^{1/\beta-1}}B. \quad (4)$$

We skip the derivations due to space constraints. From [1], the corresponding optimal hit rate and hit probability under HPB-CUM are $\lambda_i^p = \frac{w_i^{1/\beta}\lambda_i}{\sum_j w_j^{1/\beta}}B$ and $h_i^p = \frac{w_i^{1/\beta}}{\sum_j w_j^{1/\beta}}B$, respectively.

Note that $\lambda_i^r = \lambda_i^p$ and $h_i^r = h_i^p$, for $\beta = 1$, i.e., HRB-CUM and HPB-CUM are identical. Hence, we only focus on comparisons for $\beta \geq 0$ and $\beta \neq 1$ in the following sections.

3. ANALYSIS

In this section, we compare the corresponding hit probabilities h_i^r and h_i^p and hit rates λ_i^r and λ_i^p, under these two approaches for different weights. Denote $\Lambda = \sum_{i=1}^n \lambda_i$. Without loss of generality (w.l.o.g.), we assume that the arrival rates satisfy $\lambda_1 \geq \cdots \geq \lambda_n$, such that the content popularities satisfy $p_1 \geq \cdots \geq p_n$, where $p_i = \lambda_i/\Lambda$. We first consider a general request process, and then consider the Zipf distribution, i.e., $p_i = \frac{A}{i^\alpha}$ for $i = 1, \cdots, n$, satisfying $\sum_{i=1}^n p_i = 1$. In particular, we choose the Zipf parameter $\alpha = 0.8$ with $n = 10^3$ and $B = 100$ in our numerical studies.

Monotone decreasing weights: Consider the case where the weights increase with request rate. W.l.o.g., we consider monotone decreasing weights, i.e., $w_1 \geq \cdots \geq w_n$, given $\lambda_1 \geq \cdots \geq \lambda_n$. Due to space constraints, we omit all the proofs, which are available in [4].

THEOREM 1. *When weights are monotone decreasing, (i) for $\beta < 1$, HRB-CUM favors more popular items compared to HPB-CUM, i.e., $\exists j \in (1, n)$ s.t. $h_i^r > h_i^p$, $\forall i < j$, and $h_i^r < h_i^p$, $\forall i > j$; and (ii) for $\beta > 1$, HRB-CUM favors less popular items compared to HPB-CUM, i.e., $\exists l \in (1, n)$ s.t. $h_i^r < h_i^p$, $\forall i < l$, and $h_i^r > h_i^p$, $\forall i > l$. In particular, if $j, l \in \mathbb{Z}^+$, then $h_j^r = h_j^p$, and $h_l^r = h_l^p$.*

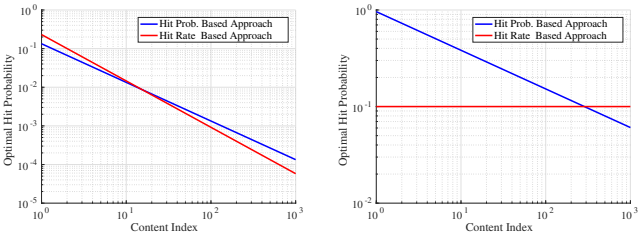

Figure 1: Hit Probability Comparison for $\beta = 0.8$ (Left) and $\beta = 2$ (Right)

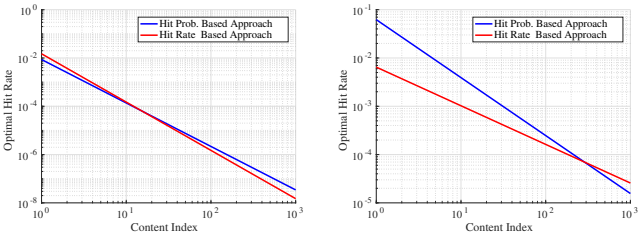

Figure 2: Hit Rate Comparison for $\beta = 0.8$ (Left) and $\beta = 2$ (Right)

The following corollary applies to the Zipf popularity distribution.

COROLLARY 1. *Under the Zipf popularity distribution, we have the following results: (a) When $\beta < 1$, $h_i^r > h_i^p$, for $i = 1, \cdots, i_0$, and $h_i^r < h_i^p$, for $i = i_0+1, \cdots, n$; (b) When $\beta > 1$, $h_i^r < h_i^p$, for $i = 1, \cdots, i_0$, and $h_i^r > h_i^p$, for $i = i_0+1, \cdots, n$, where* $i_0 = \left\lfloor \left(\frac{\sum_j w_j^{\frac{1}{\beta}} j^{\alpha(1-\frac{1}{\beta})}}{\sum_j w_j^{\frac{1}{\beta}}} \right)^{\frac{1}{\alpha(1-\frac{1}{\beta})}} \right\rfloor$.

In particular, we consider $w_i = \lambda_i$ in our numerical studies. The results are illustrated in Figure 1 for $\beta = 0.8$ and $\beta = 2$.

Now we are ready to make a comparison between the hit rate under these two approaches.

THEOREM 2. *When weights are monotone decreasing, (i) for $\beta < 1$, HRB-CUM favors more popular items compared to HPB-CUM, i.e., $\exists \tilde{j} \in (1, n)$ s.t. $\lambda_i^r > \lambda_i^p$, $\forall i < \tilde{j}$; and (ii) for $\beta > 1$, HRB-CUM favors less popular items compared to HPB-CUM, i.e., $\exists \tilde{l} \in (1, n)$ s.t. $\lambda_i^r > \lambda_i^p$, $\forall i > \tilde{l}$. In particular, if $\tilde{j}, \tilde{l} \in \mathbb{Z}^+$, then $\lambda_{\tilde{j}}^r = \lambda_{\tilde{j}}^p$, and $\lambda_{\tilde{l}}^r = \lambda_{\tilde{l}}^p$.*

The following corollary applies to the Zipf popularity distribution.

COROLLARY 2. *Under the Zipf popularity distribution, we have the following results: (a) When $\beta < 1$, $\lambda_i^r > \lambda_i^p$ for $i = 1, \cdots, i_0$, and $\lambda_i^r < \lambda_i^p$ for $i = i_0+1, \cdots, n$; (b) When $\beta > 1$, $\lambda_i^r < \lambda_i^p$ for $i = 1, \cdots, i_0$, and $\lambda_i^r > \lambda_i^p$ for $i = i_0+1, \cdots, n$, where* $i_0 = \left\lfloor \left(\frac{\sum_j w_j^{\frac{1}{\beta}} j^{\alpha(1-\frac{1}{\beta})}}{\sum_j w_j^{\frac{1}{\beta}}} \right)^{\frac{1}{\alpha(1-\frac{1}{\beta})}} \right\rfloor$.

In particular, we consider $w_i = \lambda_i$ in our numerical studies. The results are illustrated in Figure 2 for $\beta = 0.8$ and $\beta = 2$.

REMARK 1. *We also consider the case when weights are uniform, i.e., $w_1 = \cdots = w_n$, we obtain a similar threshold result between HRB-CUM and HPB-CUM. In particular, we can characterize the explicit form of the threshold i_0 under the Zipf distribution. We skip the results here due to space constraints, which are available in [4].*

4. ONLINE ALGORITHMS

In Section 2, we formulate the optimization problem with a fixed cache size, however, system parameters can change over time, and we need online algorithms to implement the optimal strategy to adapt to these changes in the presence of limited information. In the following, we develop such algorithms for HRB-CUM, and then make a comparison with the corresponding algorithms for HPB-CUM in [1]. Due to space limitations, we only focus on dual algorithms and skip the detailed proofs of all the results, which are available in [4].

Dual Algorithm: The HRB-CUM formulation in (3) is a convex optimization problem, and hence solving the dual problem will result in the optimal solution. Since we can easily ensure that $0 < t_i < \infty$, then $0 < h_i^r < 1$, i.e., $0 < \lambda_i^r < \lambda_i$. Therefore, the Lagrange dual function is given as

$$D(\eta) = \max_{\lambda_i^r} \left\{ \sum_{i=1}^{n} U_i(\lambda_i^r) - \eta \left[\sum_{i=1}^{n} \frac{\lambda_i^r}{\lambda_i} - B \right] \right\}, \quad (5)$$

and the dual problem is $\min_{\eta \geq 0} D(\eta)$. Following the standard *gradient descent algorithm* by taking the derivative of $D(\eta)$ w.r.t. η, we have $\eta \leftarrow \max \left\{ 0, \eta + \gamma \left[\sum_{i=1}^{n} \frac{\lambda_i^r}{\lambda_i} - B \right] \right\}$, where $\gamma > 0$ is the step size at each iteration and $\eta \geq 0$ due to KKT conditions.

Based on the results in Section 2, we have $\lambda_i^r = U_i'^{-1}\left(\frac{\eta}{\lambda_i}\right)$. Hit rates are then controlled by setting the timer t_i. As $\lambda_i^r = \lambda_i(1 - e^{-\lambda_i t_i})$ for a reset TTL cache, we can express t_i in terms of η, and the dual algorithm for reset TTL cache can be summarized as

$$t_i = -\frac{1}{\lambda_i} \log \left(1 - \frac{1}{\lambda_i} U_i'^{-1}\left(\frac{\eta}{\lambda_i}\right) \right), \quad (6a)$$

$$\eta \leftarrow \max \left\{ 0, \eta + \gamma \left[\sum_{i=1}^{n} \frac{\lambda_i^r}{\lambda_i} - B \right] \right\}. \quad (6b)$$

Denote the optimal value for η as η^*, then with a proper Lyapunov argument, we can show that the above dual algorithm converges to the optimal solution.

Comparisons: In the following, we implement the dual algorithm for HRB-CUM and compare it to that for HPB-CUM [1]. Due to space restrictions, we limit our study to minimum potential delay fairness, i.e., $\beta = 2$.

In our experiments, we consider a cache size $B = 100$ serving $n = 1000$ contents, where η is updated according to (6b). Requests arrive according to a Poisson process with aggregate rate $\Lambda = 1$. We assume that content popularities follow a Zipf distribution with parameter $\alpha = 0.8$.

Convergence Rate and Robustness: As discussed earlier, the dual algorithm is globally and asymptotically stable, and converges to the optimal solution. However, it says nothing about how fast it converges. From (6b), it is clear that the step size γ plays a significant role in the convergence rate. We choose different values of γ to compare the performance of HRB-CUM and HPB-CUM, shown in Figure 3. On one hand, we find that when a larger value of $\gamma = 10^{-3}$ is chosen, the dual algorithm of HRB-CUM easily converges after a few number of iterations, i.e., the simulated hit probabilities exactly match numerically computed values, while that of HPB-CUM does not converge. On the other hand,

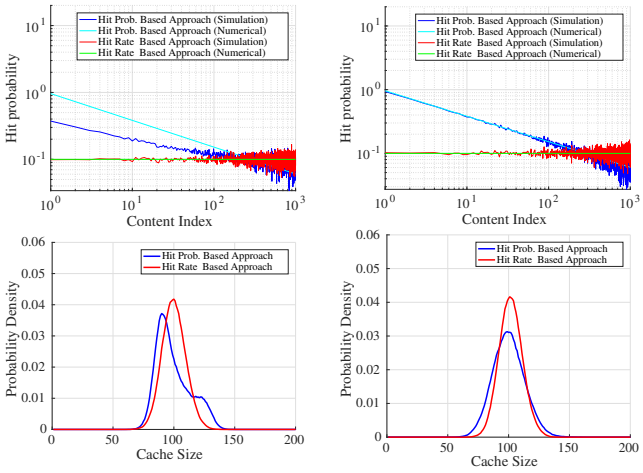

Figure 3: Hit Probability and Cache size distribution comparisons for online dual algorithm with $\gamma = 10^{-3}$ (Left) and $\gamma = 10^{-5}$ (Right)

when a smaller value $\gamma = 10^{-5}$ is chosen, both converge under the same number of iterations. We have also used $\gamma = 10^{-1}, 10^{-7}$, which exhibit similar phenomenon to 10^{-3} and 10^{-5}, respectively, and hence are omitted due to space constraints. Furthermore, we have explored the expected number of contents in the cache, shown in Figure 3. It is obvious that under the HRB-CUM, the probability of violating the target cache size B is quite small, while that probability for HPB-CUM is large especially for $\gamma = 10^{-3}$, and even for $\gamma = 10^{-5}$, HRB-CUM is more concentrated on the target size B. These results indicate that the dual algorithm associated with HRB-CUM is more robust to changes of the step size parameter γ and converge to the optimal values much faster.

REMARK 2. *We have also characterized and implemented the online primal and primal-dual algorithms. Under both cases, HRB-CUM achieves better or similar performance compared to HPB-CUM. Moreover, for proportional-fairness $\beta = 1$ and min-max fairness $\beta = \infty$, given (4), we can easily check that they are equivalent to the results in [1], hence are omitted here.*

Discussion: In this paper, we have proposed a HRB-CUM, developed decentralized online algorithms to implement the optimal policies and characterized the advantages of HRB-CUM over HPB-CUM. Further study can be done on exploring online algorithms w.r.t. estimation of content request rates. Also non-reset TTL might have different implications on the design and performance of these algorithms.

5. REFERENCES

[1] M. Dehghan, L. Massoulie, D. Towsley, D. Menasche, and Y. Tay. A Utility Optimization Approach to Network Cache Design. In *IEEE INFOCOM*, 2016.

[2] N. C. Fofack, P. Nain, G. Neglia, and D. Towsley. Analysis of TTL-based Cache Networks. In *VALUETOOLS*, 2012.

[3] N. C. Fofack, P. Nain, G. Neglia, and D. Towsley. Performance Evaluation of Hierarchical TTL-based Cache Networks. *Computer Networks*, 2014.

[4] N. K. Panigrahy, J. Li and D. Towsley. Hit Rate vs. Hit Probability for Network Cache Design. Tech. Report available at `goo.gl/O5foS0`.