Coded Caching with Partial Adaptive Matching

Jad Hachem
UCLA
jadhachem@ucla.edu

Nikhil Karamchandani
IIT Bombay
nikhilk@ee.iitb.ac.in

Sharayu Moharir
IIT Bombay
sharayum@ee.iitb.ac.in

Suhas Diggavi
UCLA
suhas@ee.ucla.edu

***Abstract*—We study the coded caching problem when we are allowed to match users to caches based on their requested files. We focus on the case where caches are divided into clusters and each user can be assigned to a unique cache from a specific cluster. We show that neither the coded delivery strategy (approximately optimal when the user-cache assignment is pre-fixed) nor the uncoded replication strategy (approximately optimal when all caches belong to a single cluster) is sufficient for all memory regimes. We propose a hybrid solution that combines ideas from both schemes and that performs at least as well as either strategy in most memory regimes. Finally, we show that this hybrid strategy is approximately optimal in most memory regimes.**

I. Introduction

Coded caching, first developed in [1], [2], uses coded broadcasts for content delivery and has been shown to provide significant benefits over traditional uncoded orthogonal delivery methods. These results have since been extended in many directions; see [3], [4] for a survey of related works. In most of these works, each user is pre-matched to a unique cache before it requests a file. On the other hand, some recent works [5], [6] have explored the possibility of actively assigning users to caches in the network, after the users have revealed their requested files, with the restriction that there be at most one user per cache. In these works, it was shown that under certain conditions on the content popularity distribution, a well-designed combination of replication, matching, and orthogonal uncoded server transmissions is sufficient for approximate optimality.

Thus, there is a dichotomy in the strategy required depending on the flexibility in user-to-cache assignment. Indeed, coded delivery is approximately optimal when the assignment is pre-fixed, while replication and uncoded delivery suffice when we have complete flexibility in the user-to-cache assignment and can match any user to any cache in the network (under the restriction of one user per cache) depending on the user demands. In this work, we explore the intermediate case, where each user can be assigned to one cache among a subset of caches. This models the possibility that, in large networks, a user will have several, but not all, caches in its vicinity, in which case it would be beneficial to match it to one of the nearby caches. As a first step in this direction, we adopt a cache-cluster model. More precisely, we make the assumption that the caches are partitioned into clusters of a fixed size, and each user can be assigned to one cache within a specific cluster, as illustrated in Fig. 1.

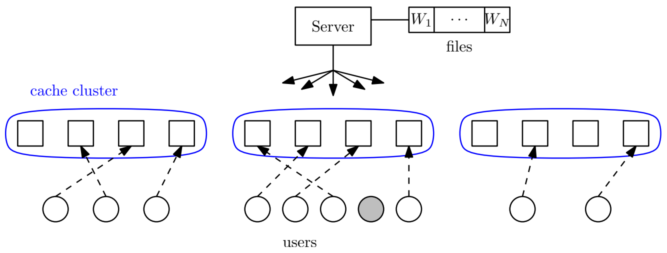

Fig. 1. Problem setup. The squares represent $K = 12$ caches, divided into clusters of $d = 4$, and the circles represent users at these clusters. Dashed arrows represent the matching phase, and the solid arrows indicate the common broadcast message. Unmatched users are in grey.

Notice that the setups in [1], [2] on the one hand and those in [5], [6] on the other hand are special cases of our setup: the former when each cluster consists of a single cache, and the latter when all caches form one cluster. Thus for each of the above strategies, we can derive a straightforward adaptation for this case with intermediate cluster size. We find that while each of these strategies is approximately optimal in individual regimes of the cache size ("memory regimes"), neither is sufficient to achieve an approximately optimal rate for all memory points. We propose a new scheme that generalizes the key ideas in both strategies and performs at least as well as they do in most memory regimes. It is a hybrid scheme that balances the possibility of matching users to caches in a cluster with the benefits of a coded delivery across clusters. We analyze the performance of this hybrid scheme and use information-theoretic outer bounds to show that it is approximately optimal in most memory regimes.

The rest of the paper is organized as follows. Section II formally defines the problem setup. We present our main results in Section III. Section IV describes the hybrid strategy in detail, while Section V develops the outer bounds to show approximate optimality. Some detailed proofs are given in the Appendix.

II. Setup

Consider the system depicted in Fig. 1. A server holds N files $W_1, \dots, W_N$ of size F bits each. There are K caches of capacity MF bits, equivalently M files, each. The caches are divided into K/d clusters of size d each, where d is assumed to divide K. For every $n \in \{1, \dots, N\}$ and every $c \in \{1, \dots, K/d\}$, there are $u_n(c)$ users accessing cluster c

This work was supported in part by NSF grant 1423271, in part by an Inspire Faculty Fellowship from the Department of Science and Technology, Government of India, and in part by a seed grant from IIT Bombay.

and requesting file W_n. We refer to the numbers $\{u_n(c)\}_{n,c}$ as the *request profile* and will often represent the request profile as a vector $\mathbf{u}$ for convenience.

In addition to the usual placement and delivery phases, there is an intermediate phase that we call the *matching phase*. The matching phase occurs before the delivery phase but after the request profile has been revealed. During the matching phase, each user is matched to a single cache *within its cluster*. If there are fewer caches than users in one cluster, then some users will be unmatched.

In this paper, we focus on the case where the numbers $u_n(c)$ are iid Poisson random variables with parameter $\rho d/N$, where $\rho \in (0, 1/4)$ is some fixed constant. The expected total number of users in the system is thus ρK. Note that this represents a uniform popularity model since the number of requests for each file is identically distributed across files.

For a given request profile $\mathbf{u}$, let $R_{\mathbf{u}}$ denote the rate of the broadcast message required to deliver to all users their requested files. For any cache memory M, our goal is to minimize the expected rate $\bar{R} = \mathbb{E}_{\mathbf{u}}[R_{\mathbf{u}}]$. Specifically, we are interested in $\bar{R}^*$ defined as the smallest $\bar{R}$ over all possible strategies. Furthermore, we assume that there are more files than caches, i.e., $N \geq K$, which is the case of most interest. We also, for analytical convenience, focus on the case where the cluster size d grows at least as fast as $\log K$. More precisely, we assume

$$d \geq \frac{2(1+\delta_0)}{\alpha} \log K, \qquad (1)$$

where $\alpha = -\log(4\rho e^{1-4\rho})$ and $\delta_0 > 0$ is some constant. Note that $\alpha > 0$.[1]

III. Main Results

Our setup generalizes two extremes that have been studied in the literature. When $d = 1$, every cluster consists of only one cache, and hence no adaptive matching is possible. This setup was studied in [1] (though with a slightly different request model[2]) and it was shown that a scheme that sends a coded message during delivery was approximately optimal; uncoded delivery was not. At the other extreme, when $d = K$, all the caches belong to one super-cluster, and any user can then be matched to any cache. For this setup, it was shown in [5] that a scheme that strategically replicates content in caches and then matches users to the cache that contains its requested file, serving them directly if they could not be matched, is approximately optimal. In fact, it turns out that a coded delivery is unnecessary for approximate optimality. Specifically, a scheme that adapts the one in [1] (by ignoring any potential of adaptive matching and instead arbitrarily matching users to caches) would perform much worse than the approximately optimal strategy for all but the smallest memory values.

To summarize, coded delivery is approximately optimal when adaptive matching is impossible ($d = 1$), but content replication with uncoded delivery is approximately optimal when maximal adaptive matching is allowed ($d = K$). Clearly, there is some transition in the suitable strategy that occurs for intermediate values of the cluster size d. This transition should balance the apparent trade-off between adaptive matching and coded delivery.

Our main contribution in this work is a strategy that generalizes themes from both the Pure Coded Delivery (PCD) scheme for $d = 1$ and the Pure Adaptive Matching (PAM) scheme for $d = K$. The idea is to apply adaptive matching at the level of *groups* of files and *groups* of caches, while performing a coded delivery within each group. This takes advantage of the flexibility offered by adaptively matching within one cluster, while still providing the benefits of coded delivery across clusters. We therefore call this strategy Hybrid Coding and Matching (HCM). The following theorem gives the expected rate achieved by HCM.

Theorem 1. *For any arbitrary $\delta \in (0, \delta_0]$, and for any $M \in [0, (8(1+\delta)N \log K)/\alpha d]$, the following expected rate is achievable:*

$$\bar{R}(M) \leq \min\left\{ \left[\frac{2N}{M} - \frac{\alpha d}{4(1+\delta)\log K}\right]^+ + \frac{K^{-\delta}}{\sqrt{2\pi}}, \rho K \right\},$$

where $[y]^+ = \max\{y, 0\}$.

Theorem 1 is proved in Section IV. Note that we can optimize over δ in order to obtain the smallest $\bar{R}(M)$.

We next compare HCM to PCD and PAM if they were adapted to this setup of intermediate values of d. By applying the same scheme from [1] with the addition of an arbitrary matching, PCD can achieve a rate similar to [1],

$$\bar{R}_{\text{PCD}} \leq \min\left\{ \frac{N}{M} - 1, \rho K \right\}.$$

As for PAM, we can apply the scheme from [5] at the cluster level to show an achievable expected rate of

$$\bar{R}_{\text{PAM}} \leq \begin{cases} \rho K & \text{if } M < O(N/d); \\ \min\left\{\rho K, KMe^{-\rho h d M/N}\right\} & \text{if } M > \Omega(N/d). \end{cases}$$

where $h = (1/\rho)\log(1/\rho) + 1 - 1/\rho$. A rough approximation of the expected rate achieved by each scheme is plotted in Fig. 2 for visualization.

Comparing PCD and PAM first, we see that PCD is the better choice for small memory while PAM is more efficient for large memory. In fact, PAM is trivially approximately optimal for $M > \Omega((N/d)\log K)$—where its rate becomes $o(1)$—while PCD is not, and PCD is approximately optimal

[1] The function $f(x) = xe^{1-x}$ is strictly increasing on the interval $(0, 1)$. Consequently, $f(x) < f(1) = 1$ for all $x \in (0, 1)$. Since $4\rho \in (0, 1)$, this implies $\alpha = -\log f(4\rho) > 0$.

[2] When d is very small, the Poisson request model adopted in this paper is not suitable. Indeed, if $d = 1$ for example, the expected number of users that cannot be matched to any cache is always a fraction of the total number of caches. This means that a rate proportional to K is inevitable even with infinite memory.

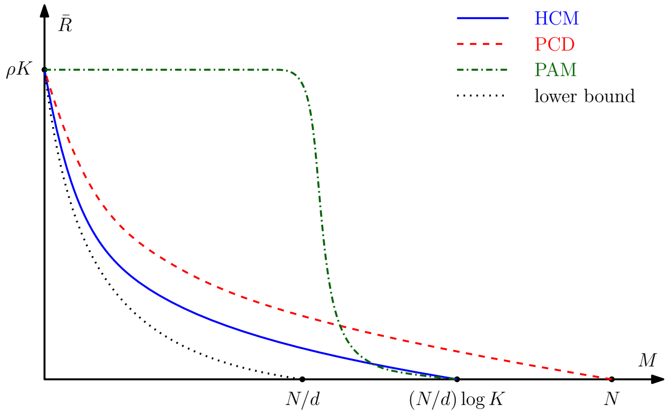

Fig. 2. The rates achieved by each scheme, as well as information-theoretic lower bounds. All values are approximate for clarity.

for $M < O(N/d)$ while PAM is not. This is consistent with our previous discussion about the two extremes: if d is very small, then most of the memory values lie in the regime where PCD is approximately optimal; if d is large, then most of the memory values lie in the regime where PAM is approximately optimal.

In this paper, we show that HCM is approximately optimal in both regimes where PCD and PAM are, respectively. Moreover, HCM is smaller than PCD by an additive gap of about $d/\log K$ for the majority of memory values, and the multiplicative gap becomes large as M approaches $(N/d)\log K$. On the other hand, HCM is order-wise better than PAM for all $M < O(N/d)$, and, similarly to PAM, it achieves a rate of $o(1)$ for $M > \Omega((N/d)\log K)$. Thus HCM is a unified scheme that generalizes the ideas of both PAM and PCD and performs at least as well as either of them in many regimes.

The approximate optimality of HCM holds trivially for $M > (8(1+\delta)N\log K)/\alpha d$, where its achieved expected rate is $K^{-\delta}/\sqrt{2\pi} = o(1)$. The following theorem states its order-optimality for smaller values of M as well.

Theorem 2. *When $N \geq 10$, the ratio of the expected rate $\bar{R}(M)$ achieved in Theorem 1 to the optimal expected rate $\bar{R}^*(M)$ is a constant C,*

$$\frac{\bar{R}(M)}{\bar{R}^*(M)} = C,$$

as long as $M \leq (1-e^{-1}/2)N/2d$. Furthermore, the rate is trivially within a constant multiplicative and additive gap from the optimum for $M \geq (8(1+\delta)N\log K)/\alpha d$.

Theorem 2 is proved in Section V. The proof uses standard cut-set bounds combined with some probability inequalities to derive lower bounds on the optimal expected rate.

IV. Hybrid Coding and Matching

In this section, we will describe the hybrid strategy that achieves the rate in the statement of Theorem 1. We will first start with a high-level overview of the scheme, and then give the details.

The idea is to divide the files into a number of colors $\chi \approx d/\log K$. For each cluster, we also divide the caches into χ colors. During the placement phase, a random sampling is done such that each cache only stores parts of files of the same color. During the matching phase, each user is matched to a cache in its cluster that has the same color as the file that it has requested. During the delivery phase, we send for every color one coded broadcast message, for a total of χ messages.

Assuming the matching phase successfully matches (almost) all users, the result is χ broadcast messages of size approximately $(N/\chi)/M - 1$ each [2], yielding a total rate of

$$\chi\left(\frac{N/\chi}{M} - 1\right) = \frac{N}{M} - \chi \approx \frac{N}{M} - \frac{d}{\log K}.$$

Note that this is the largest value of χ that we can guarantee ensures the matching phase is successful. Alternatively, the server can simply individually transmit to each user the file that they requested. Since there are ρK users on average, this gives a rate of ρK. Taking the minimum of these two quantities gives the result of Theorem 1.

We will now formalize the above argument.

Proof of Theorem 1: Let $\delta \in (0, \delta_0]$ be arbitrary, and recall that $\alpha = -\log(4\rho e^{1-4\rho}) > 0$. Choose a number of colors $\chi = \lfloor \alpha d/2(1+\delta)\log K \rfloor$. Notice that (1) ensures that $\chi \geq 1$. Divide the N files into χ groups of $\lfloor N/\chi \rfloor$ or $\lceil N/\chi \rceil$ files each.[3] Similarly, divide the d caches in each cluster into χ groups of $\lfloor d/\chi \rfloor$ each (the remaining $d - \chi\lfloor d/\chi \rfloor$ caches are ignored and never used; this makes the analysis simpler). Each group of files and each group of caches in each cluster is colored with a different color. Thus for each color, we have up to $\lceil N/\chi \rceil$ files and $(K/d)\lfloor d/\chi \rfloor$ caches.

During the placement phase, the decentralized placement strategy of [2] is used on each color individually. Specifically, each cache stores a random subset of $MF/\lfloor N/\chi \rfloor$ or $MF/\lceil N/\chi \rceil$ bits from each file of the same color.

During the matching phase, users at a cluster c requesting a file from some color x are matched with a cache from cluster c of the same color x. Excess users are left unmatched. Note that, at cluster c, the number of users requesting a file from color x is

$$U(c,x) = \sum_{n \in \mathcal{W}_x} u_n(c) \sim \text{Poisson}\left(|\mathcal{W}_x| \cdot \rho d/N\right),$$

where $\mathcal{W}_x$ is the set of files of color x, with $|\mathcal{W}_x| = \lfloor N/\chi \rfloor$ or $\lceil N/\chi \rceil$.

During the delivery phase, we choose the better of two schemes. The first scheme is to simply unicast to each user their requested file. The expected rate using this scheme is

$$\bar{R}_1 = \mathbb{E}\left[\sum_{n=1}^{N}\sum_{c=1}^{K/d} u_n(c)\right] = \mathbb{E}\left[\text{Poisson}(\rho K)\right] = \rho K. \qquad (2)$$

The second scheme is as follows. For each color, the matched users are served using a coded broadcast message

[3] Such a division is always possible.

similar to [2]. Therefore, for each color x, we have a rate of $R_2(x) \leq [|\mathcal{W}_x|/M - 1]^+$, where $[y]^+ = \max\{y, 0\}$, for a total rate of[4]

$$R_2 = \sum_{x=1}^{\chi} R_2(x) \leq \sum_{x=1}^{\chi} \left[\frac{|\mathcal{W}_x|}{M} - 1\right]^+$$
$$\leq \left[\frac{\chi \lceil N/\chi \rceil}{M} - \chi\right]^+ \leq \left[\frac{2N}{M} - \chi\right]^+. \qquad (3)$$

As for the unmatched users, they are served by unicasting to each user their requested file. If U^0 denotes the total number of unmatched users, then these users require a rate of $R_3 = U^0$. We give an upper bound on its expected value in the following lemma, proved in the Appendix.

Lemma 1. *The expected number of unmatched users can be bounded by*

$$\mathbb{E}[U^0] \leq \frac{1}{\sqrt{2\pi}} \exp\left\{\log K - \alpha \lfloor d/\chi \rfloor\right\}.$$

Because of our choice of number of colors $\chi = \lfloor \alpha d / 2(1+\delta)\log K \rfloor$, we have

$$\alpha \lfloor d/\chi \rfloor \geq \alpha d / 2\chi \geq (1+\delta)\log K.$$

Consequently, Lemma 1 implies an expected value for R_3 of

$$\bar{R}_3 = \mathbb{E}[U^0] \leq \frac{1}{\sqrt{2\pi}} \exp\{\log K - (1+\delta)\log K\} = \frac{K^{-\delta}}{\sqrt{2\pi}}. \qquad (4)$$

By combining (4) with (2) and (3), we get an achievable expected rate of

$$\bar{R} = \min\left\{\bar{R}_1, R_2 + \bar{R}_3\right\}$$
$$\leq \min\left\{\rho K, \left[\frac{2N}{M} - \chi\right]^+ + \frac{K^{-\delta}}{\sqrt{2\pi}}\right\}.$$

We have thus achieved the expected rate in Theorem 1. ■

V. Approximate Optimality

In order to prove Theorem 2, we first need to derive lower bounds on the optimal expected rate $\bar{R}^*$. These are given in the following lemma, proved at the end of the section.

Lemma 2. *Let $s \in \{1, \dots, K/d\}$. If $N \geq 10$, we have the following lower bound on $\bar{R}^*$:*

$$\bar{R}^* \geq \frac{1}{4}\rho s d \left(1 - \frac{e^{-1}}{2} - \frac{sdM}{N}\right).$$

These lower bound can be used much like in [1] in order to show approximate optimality. In fact, by writing it as

$$\bar{R}^* \geq \frac{\rho(1 - e^{-1}/2)}{4} d \cdot \max_{s \in [K/d]} s \left(1 - \frac{sdM}{(1 - e^{-1}/2)N}\right),$$

we can use the same argument as in [1] to show

$$\bar{R}^* \geq \frac{\rho(1 - e^{-1}/2)}{4} d \cdot \frac{1}{12} \min\left\{\frac{(1 - e^{-1}/2)N}{dM} - 1, \frac{K}{d}\right\}$$
$$= \frac{\rho(1 - e^{-1}/2)}{48} \min\left\{\frac{(1 - e^{-1}/2)N}{M} - d, K\right\}.$$

This matches the achievable rate up to a constant in the regime $M < (1 - e^{-1}/2)N/2d$.

Proof of Lemma 2: First, consider the following hypothetical scenario. Let there be a single cache of size M', and suppose that a request profile $\mathbf{u}$ is issued from users all connected to this one cache. Moreover, assume that we allow the designer to set the cache contents *after* the request profile is revealed; thus both the placement and delivery take place with knowledge of $\mathbf{u}$. If we send a single message to serve those requests, and denote its rate by $R'(M', \mathbf{u})$, then a cut-set bound shows that

$$R'(M', \mathbf{u}) + M' \geq \gamma(\mathbf{u}), \qquad (5)$$

where $\gamma(\mathbf{u})$ is the total number of *distinct* files requested in $\mathbf{u}$.

Since all users share the same resources, if a file can be decoded by one user then it can be decoded by all. Thus the number of requests for each file is irrelevant for (5), as long as it is non-zero. Furthermore, since both the placement and the delivery are made after the request profile is revealed, the *identity* of the requested files is irrelevant. Indeed, if $R'(M', \mathbf{u}_1) > R'(M', \mathbf{u}_2)$ for some $\mathbf{u}_1$ and $\mathbf{u}_2$ such that $\gamma(\mathbf{u}_1) = \gamma(\mathbf{u}_2)$, then a simple relabling of the files in $\mathbf{u}_1$ can make it equivalent to $\mathbf{u}_2$, and thus the same rate can be achieved. Consequently, (5) can be rephrased using only the *number* of distinct requested files,

$$\widetilde{R}(M', y) + M' \geq y, \qquad (6)$$

where $\widetilde{R}(M', y)$ is the rate required to serve requests for y distinct files from one cache of memory M'. Note that $R'(M', \mathbf{u}) = \widetilde{R}(M', \gamma(\mathbf{u}))$ for all $\mathbf{u}$. Additionally, it can be seen that $\widetilde{R}(M', y)$ increases as y increases: if $y_1 < y_2$, then we can always add $y_2 - y_1$ users to request new files, and thus achieve a rate of $\widetilde{R}(M', y_1) \leq \widetilde{R}(M', y_2)$.

Let us now get back to our original problem. For convenience, define $\bar{R}_y = \mathbb{E}_{\mathbf{u}}[R_{\mathbf{u}} | \gamma(\mathbf{u}) = y]$ for every y. Suppose we choose $s \in \{1, \dots, K/d\}$ different clusters, and we observe the system over B instances. Over this period, a certain number of users will connect to these clusters and request files; all other users are ignored. If we denote the resulting request profiles as $\mathbf{u}_1, \dots, \mathbf{u}_B$, then the rate required to serve all requests is $\sum_{i=1}^{B} R_{\mathbf{u}_i}$.

Suppose we relax the problem and allow the users to co-operate. Suppose also that we allow the placement to take place after all B request profiles are made. This can only reduce the required rate. Furthermore, this is now an equivalent problem to the hypothetical scenario described at the beginning of the proof. Therefore, if we denote by $\bar{\mathbf{u}}^B$ the request profile cumulating $\mathbf{u}_1, \dots, \mathbf{u}_B$, we have

$$\sum_{i=1}^{B} R_{\mathbf{u}_i} \geq R'(sdM, \bar{\mathbf{u}}^B) = \widetilde{R}(sdM, \gamma(\bar{\mathbf{u}}^B)).$$

[4] To be precise, we get $R_2 \leq N/M - \chi$ for $M \leq \lfloor N/\chi \rfloor$ and $R_2 = 0$ for $M \geq \lceil N/\chi \rceil$. However, we will use the upper bound in (3) for simplicity.

By averaging over the cumulative request profile, we obtain the following bound on any achievable expected rate $\bar{R}$:

$$\begin{aligned} B\bar{R} &\geq \mathbb{E}_{\bar{\mathbf{u}}^B}\left[\widetilde{R}(sdM,\gamma(\bar{\mathbf{u}}^B))\right] \\ &= \mathbb{E}_{Y_{sB}}\left[\mathbb{E}_{\bar{\mathbf{u}}^B}\left[\widetilde{R}(sdM,\gamma(\bar{\mathbf{u}}^B))\middle|\gamma(\bar{\mathbf{u}}^B)=Y_{sB}\right]\right] \\ &= \mathbb{E}_{Y_{sB}}\left[\widetilde{R}(sdM,Y_{sB})\right], \end{aligned} \tag{7}$$

where Y_{sB} is a random variable denoting the number of distinct files requested after B instances at s clusters. Since (7) holds for all achievable rates $\bar{R}$, it also holds for $\bar{R}^*$.

Let us choose $B = \lceil N/\rho sd \rceil$. Using Chernoff bounds, we obtain some probabilistic bounds on the number of requested files Y_{sB}. These bounds are given in the following lemma for convenience; the lemma is proved in the Appendix.

Lemma 3. *If Y denotes the number of distinct requested files by users at s clusters over $B = \lceil N/\rho sd \rceil$ instances, then, for all $\epsilon > 0$,*

$$\Pr\left\{Y \leq (1-e^{-1}-\epsilon)N\right\} \leq e^{-ND(e^{-1}+\epsilon||e^{-1})}.$$

We will now use Lemma 3 to obtain bounds on the expected rate. Define $\widetilde{N} = (1-e^{-1}-\epsilon)N$. From (7), we have

$$\begin{aligned} \lceil N/\rho sd \rceil\,\bar{R}^* &\geq \mathbb{E}_{Y_{sB}}\left[\widetilde{R}(sdM,Y_{sB})\right] \\ &= \sum_{y=1}^{N} \Pr\{Y_{sB}=y\}\cdot\widetilde{R}(sdM,y) \\ &\geq \sum_{y=\widetilde{N}}^{N} \Pr\{Y_{sB}=y\}\cdot\widetilde{R}(sdM,y) \\ &\overset{(a)}{\geq} \widetilde{R}(sdM,\widetilde{N})\sum_{y=\widetilde{N}}^{N}\Pr\{Y=y\} \\ &\overset{(b)}{\geq} \widetilde{R}(sdM,\widetilde{N})\left(1-e^{-ND(e^{-1}+\epsilon||e^{-1})}\right) \\ &\overset{(c)}{\geq} (1-o(1))\left((1-e^{-1}-\epsilon)N-sdM\right), \end{aligned}$$

where (a) uses the fact that $\widetilde{R}(sdM,y)$ can only increase with the number of requested files y, (b) follows from Lemma 3, and (c) is due to (6).

For a fixed ϵ, the $1-o(1)$ factor approaches 1 as N grows. More generally, we can lower-bound it by some constant for a large enough N. For example, if $\epsilon = e^{-1}/2$, then the term is larger than $1/2$ as long as $N \geq 10$. Therefore, we have

$$\bar{R}^* \geq \frac{\left(1-\frac{e^{-1}}{2}\right)N - sdM}{2\lceil N/\rho sd\rceil} \geq \frac{1}{4}\rho sd\left(1-\frac{e^{-1}}{2}-\frac{sdM}{N}\right)$$

as long as $N \geq 10$. More generally, we can approach

$$\bar{R}^* \geq \frac{\left(1-e^{-1}\right)N - sdM}{\lceil N/\rho sd\rceil} \geq \frac{1}{2}\rho sd\left(1-e^{-1}-\frac{sdM}{N}\right),$$

if we allow N to be sufficiently large. ■

References

[1] M. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[2] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1029–1040, Aug 2015.

[3] ——, "Coding for caching: fundamental limits and practical challenges," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 23–29, August 2016.

[4] G. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, "Wireless caching: technical misconceptions and business barriers," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 16–22, August 2016.

[5] M. Leconte, M. Lelarge, and L. Massoulié, "Bipartite graph structures for efficient balancing of heterogeneous loads," *SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 1, pp. 41–52, Jun. 2012.

[6] S. Moharir and N. Karamchandani, "Content replication in large distributed caches," *arXiv:1603.09153 [cs.NI]*, Mar. 2016.

Appendix

Proof of Lemma 1: Let $V(c,x)$ denote the number of unmatched users for color x in cluster c, i.e., $V(c,x) = 0$ if $U(c,x) \leq \lfloor d/\chi \rfloor$ and $V(c,x) = U(c,x) - \lfloor d/\chi \rfloor$ otherwise. Therefore,

$$U^0 = \sum_{c,x} V(c,x) \implies \mathbb{E}[U^0] = \sum_{c,x}\mathbb{E}[V(c,x)].$$

Before we proceed, it will be useful to state the following two properties of Poisson variables, proved at the end of the Appendix.

Proposition 1. *If Y is a Poisson variable with parameter λ, then for all integers $m \geq 1$,*

$$\mathbb{E}\left[Y|Y \geq m\right] = m\Pr\{Y=m|Y\geq m\} + \lambda.$$

Proposition 2. *If Y is a Poisson variable with parameter λ, then, for all integers $m \geq 1$, the function $\lambda \mapsto \Pr\{Y=m\}$ is increasing in λ as long as $\lambda < m$.*

Let us write $V = V(c,x)$, $U = U(c,x)$, and $q = \lfloor d/\chi \rfloor$ for simplicity. Using the tower property,

$$\begin{aligned} \mathbb{E}\left[V\right] &= \mathbb{E}\left[\mathbb{E}\left[V|U\right]\right] \\ &= \Pr\left\{U<q\right\}\cdot\mathbb{E}\left[V|U<q\right] \\ &\quad + \Pr\left\{U\geq q\right\}\cdot\mathbb{E}\left[V|U\geq q\right] \\ &= 0 + \Pr\left\{U\geq q\right\}\cdot\mathbb{E}\left[U-q|U\geq q\right] \\ &= \Pr\left\{U\geq q\right\}\cdot\left(\mathbb{E}\left[U|U\geq q\right]-q\right) \\ &\overset{(a)}{=} \Pr\left\{U\geq q\right\}\cdot q\Pr\left\{U=q|U\geq q\right\} \\ &\quad + \left(|\mathcal{W}_x|\rho d/N - q\right)\Pr\left\{U\geq q\right\} \\ &= q\cdot\Pr\left\{U=q\right\} - \left(q-\rho|\mathcal{W}_x|d/N\right)\cdot\Pr\left\{U\geq q\right\} \\ &\overset{(b)}{\leq} q\cdot\Pr\left\{U=q\right\}, \end{aligned}$$

where (a) uses Proposition 1 and the fact that $U = U(c,x)$ is a Poisson variable with parameter $|\mathcal{W}_x|\rho d/N$, and (b) follows from

$$\rho|\mathcal{W}_x|d/N \leq \rho\left\lceil N/\chi\right\rceil d/N \leq 2\rho d/\chi \leq 4\rho\left\lfloor d/\chi\right\rfloor, \tag{8}$$

which, combined with $4\rho < 1$, gives us $q = \lfloor d/\chi \rfloor > \rho|\mathcal{W}_x|d/N$. Therefore,

$$\mathbb{E}[V(c,x)] \leq \lfloor d/\chi \rfloor \cdot \Pr\left\{U(c,x) = \lfloor d/\chi \rfloor\right\}. \tag{9}$$

Let $\widetilde{U}$ be a Poisson variable with parameter $4\rho\left\lfloor d/\chi \right\rfloor$. Recall that $U(c,x)$ is a Poisson variable with parameter $\rho|\mathcal{W}_x|d/N$, which is less than $4\rho\left\lfloor d/\chi \right\rfloor$ by (8). Since we thus have

$$\rho|\mathcal{W}_x|d/N \le 4\rho\left\lfloor d/\chi \right\rfloor < \left\lfloor d/\chi \right\rfloor,$$

we can apply Proposition 2 on (9) to obtain

$$\mathbb{E}[V(c,x)] \le \left\lfloor d/\chi \right\rfloor \cdot \Pr\{\widetilde{U} = \left\lfloor d/\chi \right\rfloor\}$$

for all c and x. Consequently,

$$\begin{aligned}
\mathbb{E}[U^0] &= \sum_{c,x} \mathbb{E}[V(c,x)] \\
&\le \sum_{c,x} \left\lfloor d/\chi \right\rfloor \cdot \Pr\{\widetilde{U} = \left\lfloor d/\chi \right\rfloor\} \\
&= \sum_{c,x} \left\lfloor d/\chi \right\rfloor \cdot \frac{\left(4\rho\left\lfloor d/\chi \right\rfloor\right)^{\left\lfloor d/\chi \right\rfloor} e^{-4\rho\left\lfloor d/\chi \right\rfloor}}{\left\lfloor d/\chi \right\rfloor!} \\
&= \frac{K}{d} \cdot \chi \cdot \left\lfloor d/\chi \right\rfloor \cdot \frac{(4\rho q)^q\, e^{-4\rho q}}{q!} \\
&\le K \cdot \frac{(4\rho q)^q e^{-4\rho q}}{q!},
\end{aligned}$$

where we have used $q = \left\lfloor d/\chi \right\rfloor$ again. By Stirling's approximation, we have

$$q! \ge \sqrt{2\pi} q^{q+\frac{1}{2}} e^{-q},$$

yielding

$$\begin{aligned}
\mathbb{E}[U^0] &\le \frac{1}{\sqrt{2\pi}} \cdot K \cdot \frac{(4\rho q)^q e^{-4\rho q}}{q^{q+\frac{1}{2}} e^{-q}} \\
&= \frac{1}{\sqrt{2\pi}} \cdot K \cdot \left(\frac{4\rho q}{q}\right)^q \cdot q^{-\frac{1}{2}} \cdot e^{-4\rho q + q} \\
&\le \frac{1}{\sqrt{2\pi}} \cdot K \cdot (4\rho)^q \cdot e^{(1-4\rho)q} \\
&= \frac{1}{\sqrt{2\pi}} \cdot \exp\left\{\log K + q\log 4\rho + (1-4\rho)q\right\} \\
&= \frac{1}{\sqrt{2\pi}} \cdot \exp\left\{\log K - \left(-\log 4\rho - (1-4\rho)\right)q\right\} \\
&= \frac{1}{\sqrt{2\pi}} \cdot \exp\left\{\log K - \alpha\left\lfloor \frac{d}{\chi} \right\rfloor\right\}.
\end{aligned}$$

This concludes the proof of the lemma. ■

Proof of Lemma 3: Recall that we are considering s clusters and $B = \lceil N/\rho s d \rceil$ instances of the problem. Let U_n denote the number of requests for file n by users in these clusters across the B instances. We can see that U_n is a Poisson variable with parameter

$$\lambda = \frac{\rho d}{N} \cdot sB = \frac{\rho s d}{N} \cdot \left\lceil \frac{N}{\rho s d} \right\rceil.$$

Note that $\lambda \ge 1$.

Let Z_n be equal to one if $U_n \ge 1$, i.e., if file n was requested at least once, and equal to zero otherwise. Thus Z_n is a Bernoulli variable with parameter $\Pr\{U_n \ge 1\} = 1 - e^{-\lambda}$. Then, the total number of distinct requested files can be written as $Y = \sum_n Z_n$.

Let $\epsilon > 0$ be arbitrary, and define $\eta = 1 - e^{-1} - \epsilon$. We can now use the Chernoff bound to write, for every $t > 0$,

$$\Pr\left\{Y \le \eta N\right\} \le e^{t\eta N} \cdot \mathbb{E}\left[e^{-tY}\right] = e^{t\eta N} \cdot \prod_{n=1}^{N} \mathbb{E}\left[e^{-tZ_n}\right].$$

The expression inside the product is

$$\begin{aligned}
\mathbb{E}\left[e^{-tZ_n}\right] &= e^{-\lambda} \cdot 1 + (1-e^{-\lambda}) \cdot e^{-t} = e^{-\lambda}(1-e^{-t}) + e^{-t} \\
&\le e^{-1}(1-e^{-t}) + e^{-t} = e^{-1} + e^{-t}(1-e^{-1}),
\end{aligned}$$

where the inequality is due to $\lambda \ge 1$ and the fact that the function $\lambda \mapsto e^{-\lambda}(1-e^{-t})$ decreases as λ increases, for all $t > 0$. Consequently,

$$\Pr\left\{Y \le \eta N\right\} \le \left[e^{t\eta} \cdot \left(e^{-1} + e^{-t}(1-e^{-1})\right)\right]^N.$$

By choosing t such that

$$e^t = \frac{e^{-1}+\epsilon}{e^{-1}} \cdot \frac{1-e^{-1}}{1-(e^{-1}+\epsilon)},$$

we get

$$\Pr\left\{Y \le \eta N\right\} \le \left[e^{-D(e^{-1}+\epsilon||e^{-1})}\right]^N = e^{-ND(e^{-1}+\epsilon||e^{-1})},$$

which concludes the proof. ■

Proof of Proposition 1: First notice that

$$\begin{aligned}
\lambda \Pr\{Y = m-1\} &= \lambda \cdot \frac{\lambda^{m-1} e^{-\lambda}}{(m-1)!} \\
&= \frac{\lambda^m e^{-\lambda}}{m!} \cdot m \\
&= m \Pr\{Y = m\}.
\end{aligned} \tag{10}$$

We can now write the conditional expectation as

$$\begin{aligned}
\mathbb{E}\left[Y|Y \ge m\right] &= \sum_{y=m}^{\infty} y \cdot \frac{\Pr\{Y=y\}}{\Pr\{Y \ge m\}} \\
&= \frac{1}{\Pr\{Y \ge m\}} \sum_{y=m}^{\infty} y \cdot \frac{\lambda^y e^{-\lambda}}{y!} \\
&= \frac{1}{\Pr\{Y \ge m\}} \lambda \sum_{y=m}^{\infty} \frac{\lambda^{y-1} e^{-\lambda}}{(y-1)!} \\
&= \frac{1}{\Pr\{Y \ge m\}} \lambda \cdot \Pr\{Y \ge m-1\} \\
&= \frac{\lambda \Pr\{Y = m-1\} + \lambda \Pr\{Y \ge m\}}{\Pr\{Y \ge m\}} \\
&\overset{(a)}{=} \frac{m \Pr\{Y = m\} + \lambda \Pr\{Y \ge m\}}{\Pr\{Y \ge m\}} \\
&= m \Pr\{Y = m | Y \ge m\} + \lambda,
\end{aligned}$$

where (a) uses (10) ■

Proof of Proposition 2: Define $f_m(\lambda) = \Pr\{Y = m\}$ when Y is Poisson with parameter λ, i.e., $f_m(\lambda) = \lambda^m e^{-\lambda}/m!$. Then,

$$\begin{aligned}
f'_m(y) &= \frac{1}{m!}\left(m\lambda^{m-1}e^{-\lambda} - \lambda^m e^{-\lambda}\right) \\
&= \frac{\lambda^{m-1}e^{-\lambda}}{m!}\left(m-\lambda\right).
\end{aligned}$$

Consequently, $f'_m(y) > 0$ if and only if $\lambda < m$, and hence $\Pr\{Y = m\}$ increases with λ as long as $\lambda < m$. ■