Effects of Storage Heterogeneity in Distributed Cache Systems

Kota Srinivas Reddy, Sharayu Moharir and Nikhil Karamchandani
Department of Electrical Engineering, Indian Institute of Technology Bombay
Email: ksreddy@ee.iitb.ac.in, sharayum@ee.iitb.ac.in, nikhilk@ee.iitb.ac.in

***Abstract*—In this work, we focus on distributed cache systems with non-uniform storage capacity across caches. We compare the performance of our system with the performance of a system with the same cumulative storage distributed evenly across the caches. We characterize the extent to which the performance of the distributed cache system deteriorates due to storage heterogeneity. The key takeaway from this work is that the effects of heterogeneity in the storage capabilities depend heavily on the popularity profile of the contents being cached and delivered. We analytically show that compared to the case where contents popularity is comparable across contents, lopsided popularity profiles are more tolerant to heterogeneity in storage capabilities. We validate our theoretical results via simulations.**

I. Introduction

Recent Internet usage patterns show that Video on Demand (VoD) services, e.g., YouTube [1] and Netflix [2], account for ever-increasing fractions of Internet traffic [3]. To meet the increasing demand, most popular VoD services use content delivery networks (CDNs). We focus on multiple geographically co-located caches, each with limited storage and service capabilities, deployed to serve users in that area. The motivation behind deploying local caches is to serve most user requests locally. Requests that can't be served locally are served by a central server (which stores the entire content catalog) via a root node, see Figure 1. This setting, also studied in [4], models networks where, *(i)* the ISP (root node) uses local caches to reduce the load on the network backbone or *(ii)* this geographically co-located cache cluster is a part of a larger tree network [5].

Most VoD service offer catalogs consisting of a large number of contents and serve a large number of users. Motivated by this, we study a time-slotted setting where a batch of requests arrive in each time-slot and every cache can serve at most one request in a batch. Requests that cannot be served locally by the caches are assigned to the central server. Storage and service policies are designed to minimize the number of contents which need to be fetched from the central server to serve all the requests in a batch.

The existing body of work in this space considers the setting where storage capabilities are uniform across caches [4], [6].

This work was supported in part by the Bharti Centre for Communication at IIT Bombay. The work of Sharayu Moharir and Nikhil Karamchandani was supported in part by seed grants from IIT Bombay and an Indo-French grant on "Machine Learning for Network Analytics". The work of Nikhil Karamchandani was also supported in part by the INSPIRE Faculty Fellowship from the Govt. of India.

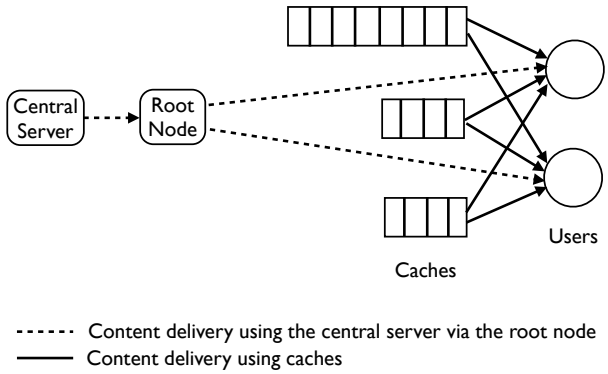

Fig. 1. *An illustration of a cache cluster consisting of three caches serving two users. The first cache has more storage than the other two. Each user can either be served by the caches or by the central server via the root node.*

In this work, we study the effects of heterogeneity in storage across caches on the performance of the system. The key takeaway of this work is that the effect of heterogeneity in storage capabilities across caches depends on the popularity profile of the contents. We show that as content popularity becomes more lopsided, the system can handle more heterogeneity in cache storage capabilities, i.e., for the same amount of cumulative memory, the performance of the heterogeneous system remains comparable to the performance of a system with uniform storage across caches.

Intuitively our results can be explained as follows. Increasing the number of contents stored on a cache increases the utility of that cache as it can be used to serve a request for any one of the stored contents. When content popularity is comparable across contents, the fraction of requests in a batch for any particular content is small. As a result, for a cache with limited storage, it is likely that none of the stored contents are requested, thus leaving the cache unutilized. This increases the number of requests that have to be served via the central server. In contrast, when content popularity is lopsided, the caches with limited storage can be used to store and serve requests for popular contents and the caches with large storage can store a mixture of some popular and a larger number of unpopular contents. This ensures that most caches are utilized, thus reducing the number of requests served centrally.

The main focus of this work is to study the impact of heterogeneity in storage sizes on the performance of a single-layer distributed caching system with a central server. This

aspect has been addressed in some other settings as well. [7] models a caching network as a graph with a cache at each vertex and explores sizing the individual caches according to various vertex centrality metrics. [8] studies a multi-tier caching network, with a possibly different cache size at each layer. The setting where each user is pre-matched to a server and the central server communicates with the users via an error-free broadcast link has been studied recently under the moniker *'coded caching'* in [9] and the impact of heterogeneity in cache sizes in this setting has been explored in [10]–[12].

A. Contributions

The main contributions of this work can be summarized as follows:

1) We first consider the case where the content popularity distribution follows the Zipf distribution (defined in Section II-B) with parameter less than 1, which corresponds to the case where the popularity is comparable across contents. We show that even if a constant fraction of the caches are restricted to have small memory size as compared to the remaining caches, the required expected server transmission rate can be much larger than a homogeneous system with the same cumulative memory.
2) Next, we consider the case where the content popularity distribution follows the Zipf distribution with parameter larger than 1 and as a result, the content popularity is more lopsided. Unlike the previous case, even if all the memory is concentrated in only a vanishing fraction of the caches, the performance of the system will be similar to a homogeneous system with the same cumulative memory.

The above results suggest that caching systems are more tolerant to heterogeneity in storage under content popularity distributions which are more lopsided than when popularity is comparable across contents.

II. SETTING

We study a system consisting of a central server, and m co-located caches, each with limited storage and service capabilities. The central server stores n files[2] of equal size (say 1 unit = b bits), where $n = \Theta(m^\gamma)$, for some $\gamma \geq 1$. Users make requests for these files, and the user requests are served using the caches and the central server.

The system operates in two phases: the first phase is the *placement phase,* in which each cache stores content related to the n files and the next phase is the *delivery phase,* in which a batch of requests arrives and are served by the caches and the central server. While files can be split for storage and transmission, this work is restricted to uncoded policies during the placement and the delivery phases. We study the asymptotic performance of the system as n, $m \to \infty$.

[2]We use the terms 'content' and 'file' interchangeably.

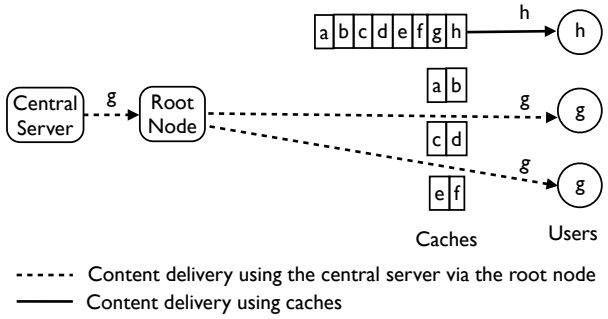

Fig. 2. *An illustration of a system consisting of four caches serving three users. The catalog consists of eight files $\{a, b, c, d, e, f, g, h\}$. The first user requests file h, while the other two request file g. The first user is served by the first cache. The other two users are served by the central server. Since both users request for the same file g, the central server sends file g to the root node, therefore, the transmission rate in this example is one.*

A. Storage Model

Cache i has the capacity to store k_i units of data. Let $M = \sum_{i=1}^{m} k_i$ denote the cumulative cache memory. Without loss of generality, we assume caches are arranged in decreasing order of storage capacity, i.e., if $i < j$, then $k_i \geq k_j$.

B. Request Model

We assume a time-slotted system. In each time-slot, a batch of $\widetilde{m} = \rho m$ (for some $\rho < 1$) requests arrive from users according to an i.i.d. distribution. Files are indexed in decreasing order of popularity.

Numerous empirical studies have shown that content popularity in VoD services follows the Zipf's law [13]–[16]. Zipf's law states that the popularity of the i^{th} most popular content is proportional to $i^{-\beta}$, where β is a positive constant known as the Zipf parameter. Small values of β imply that content popularity is comparable across contents while larger values of β correspond to lopsided popularity distributions. Typical values of β lie between 0.6 and 2.

C. Service Model

We assume a delay-intolerant uncoded service system, i.e., all user requests in a given time slot have to be served jointly by the caches and the central server in that time-slot without queuing and coding. To begin with, depending on user requests, we match users with the caches such that no cache is matched[3] to more than 1 user. Depending on the user requests and the matching between the users and the caches, the central server then transmits a message to the root node which then relays it directly to the users. Using the data received from the assigned caches and the central server message, each user should be able to reconstruct the requested file. Refer to Figure 2 for an example.

[3]The more general setting where each cache can serve upto $a \geq 1$ requests simultaneously has been analyzed in [6] for the case of homogeneous cache sizes. A similar analysis can be attempted for the case of heterogeneous cache sizes, however we do not pursue that direction in this paper.

D. Goal

The reason for deploying local caches is that they can help reduce the load on the bottleneck link between the central server and the root node. The goal in such systems is to design efficient storage and service policies to reduce the expected transmission rate required from the central server to satisfy all user requests, where the expectation is with respect to the file popularity distribution. Note that if a content needs to be delivered by the central server to multiple users in a time-slot, the central server transmits it to the root node only once. Our storage and service policies depend on the file popularity distribution.

In a departure from the existing body of work on content caching/delivery policies, we characterize the performance of various caching policies for the setting where storage is heterogeneous across caches.

III. Main results and discussion

In this section, we state and discuss our main results. Proofs are given in Section V.

We study distributed cache systems characterized as follows:

Assumption 1 (Distributed Cache System):

- m caches.
- $n = m^{\gamma}$ files for $\gamma \geq 1$.
- All files are of equal size, normalized to one unit.
- File popularity: Zipf distribution with parameter β.
- Cumulative cache memory is M units, where $M = m^{\mu}$ for $\mu \geq 1$.
- Each cache can store at least one full file, i.e., $k_i \geq 1$ $\forall\ i$.
- Requests are received in batches of $\widetilde{m} = \rho m$, where $\rho < 1$. Each request is generated i.i.d. according to the popularity distribution.
- At most one request in a batch can be allocated to each cache.

A. Zipf distribution with $\beta \in [0, 1)$

We first characterize the performance of a distributed cache system when file popularity follows the Zipf distribution with parameter $\beta \in [0, 1)$.

In addition to understanding the fundamental limit on the performance of any policy, we also evaluate the performance of a policy called Proportional Placement and Maximum Matching (PPMM) proposed in [17]. [6, Theorem 1] characterizes the performance of the PPMM policy for a homogeneous cache system with the number of caches scaling linearly with the number of files. In the PPMM policy, the number of caches that store copies of a file are proportional to its popularity. File copies are stored on caches such that no cache stores the same file multiple times. Once a batch of requests is revealed, a bipartite graph $G(V_1, V_2, E)$ is created, where V_1 is the set of requests, V_2 is the set of caches, and E is the set of edges. There is an edge between $v_1 \in V_1$ and $v_2 \in V_2$ if Cache v_2 can serve request v_1, *i.e.*, if it stores a copy of the requested file. Once the bipartite graph is created, the maximum cardinality matching between the set of requests (V_1) and the set of caches (V_2) is found. All the matched requests are served by the corresponding caches and all the unmatched requests are served by the central server via the root-node. Note that this policy satisfies our service constraint that no cache is allocated more than one request in a batch.

The following result is a straightforward generalization of [6, Theorem 1] and characterizes the performance of the PPMM policy for a homogeneous cache system, i.e., a system where all caches have the same storage capabilities.

Theorem 1: Consider a homogeneous distributed cache system satisfying Assumption 1 where all caches have equal storage capacity of M/m units and file popularity follows the Zipf distribution with parameter $\beta \in [0, 1)$. For this system, let $R^{\text{PPMM}}_{z_{[0,1)}}$ be the central server's transmission rate for the PPMM policy described above. Then, we have that,

$$\mathbb{E}\left[R^{\text{PPMM}}_{z_{[0,1)}}\right] = \begin{cases} \mathrm{O}(m) & \text{if } M < (1-\epsilon)n,\ \epsilon > 0, \\ \mathrm{O}\big(m^2 e^{-\frac{M}{n}}\big) & \text{if } M \geq n. \end{cases}$$

We use this result to characterize the amount of memory needed in a homogeneous system to ensure that for the PPMM policy, the expected transmission rate of the central server goes to zero as the system size m scales.

Corollary 1: Consider a homogeneous distributed cache system satisfying Assumption 1 where all caches have equal storage capacity of M/m units and file popularity follows the Zipf distribution with parameter $\beta \in [0, 1)$. For this system, let $R^{\text{PPMM}}_{z_{[0,1)}}$ be the central server's transmission rate for the PPMM policy described above. If $M \geq 3n \ln m = \Omega\big(n \ln m\big)$, $\mathbb{E}\big[R^{\text{PPMM}}_{z_{[0,1)}}\big] = \mathrm{o}(1)$.

We thus conclude that, for a homogeneous distributed cache system and the PPMM policy, a cumulative cache memory of $M = \Omega(n \ln m)$ is sufficient to ensure that the expected transmission rate of the central server goes to zero as the system size m scales.

Our next result focuses on a heterogeneous distributed cache system, i.e., a distributed cache system where storage is non-uniform across caches. It characterizes the fundamental limit on the performance of any policy and evaluates the performance of the PPMM policy for such a system.

Theorem 2: Consider a heterogeneous distributed cache system satisfying Assumption 1 where file popularity follows the Zipf distribution with parameter $\beta \in [0, 1)$.

(a) *Lower bound on transmission rate*: Let $\widetilde{R}^{*}_{z_{[0,1)}}$ be the central server's transmission rate for optimal policy. There exists an $\alpha(\rho) \in (0, 1)$ such that if a fraction of the m caches, say $m_2 = \alpha \cdot m$ caches, have at most $\mathrm{O}\big(n/m^{\frac{1}{1-\beta}}\big)$ units of memory, then, $\mathbb{E}\big[\widetilde{R}^{*}_{z_{[0,1)}}\big] = \omega(1)$.

(b) *Performance of PPMM*: Let $\widetilde{R}^{\text{PPMM}}_{z_{[0,1)}}$ be the central server's transmission rate for the PPMM policy described above. Then for any $c > 0$ and $\delta < 1$, if a fraction of the m caches, say $m_1 = (\rho + c)m$ caches, have at least $\Omega(n/m^{\delta})$ units of memory, then, $\mathbb{E}\big[\widetilde{R}^{\text{PPMM}}_{z_{[0,1)}}\big] = \mathrm{o}(1)$.

Intuitively, the lower bound on the transmission rate for any policy can be explained as follows. Increasing the number of contents stored on a cache increases the potential utility of that cache as it can be used to serve a request for any one of the stored contents. When storage is non-uniform across caches and content popularity is comparable across files, the utility of caches with limited storage capabilities is small since content popularity being comparable across files ensures that the fraction of requests for any particular content is small. A consequence of this is that it is very likely that many caches with limited memory go unutilized when serving a batch of requests. A large number of unutilized caches is equivalent to a large number of requests being served via the central server, thus increasing its transmission rate.

In the next result, we use Theorem 2(a) and Corollary 1 to highlight the difference between homogeneous and heterogeneous cache systems with the same cumulative memory.

Corollary 2: Let $\alpha \in (0,1)$ be as defined in Theorem 2(a) and $\delta > 0$. Consider distributed caching systems satisfying Assumption 1 where file popularity follows the Zipf distribution with parameter $\beta \in [0,1)$. Consider a heterogeneous system such that $\alpha \cdot m$ caches each have at most $\mathrm{O}\big(n/m^{\frac{1}{1-\beta}}\big)$ units of memory and the remaining $(1-\alpha)m$ caches each have memory $\Theta\big(n/m^{1-\delta}\big)$. Then, $\mathbb{E}\big[\widetilde{R}^*_{z_{[0,1)}}\big] = \omega(1)$.

On the other hand, for a homogeneous system with the same cumulative cache memory $M = \Theta\big(n \cdot m^{\delta}\big)$ as the above heterogeneous system, we have $\mathbb{E}\big[R^{\text{PPMM}}_{z_{[0,1)}}\big] = \mathrm{o}(1)$.

B. Zipf distribution with $\beta > 1$

We now compare the performances of homogeneous and heterogeneous systems when file popularity follows the Zipf distribution with parameter $\beta > 1$. The next result provides lower bounds on the expected transmission rate for a system with cumulative cache storage of M units.

Theorem 3: Consider a distributed cache system satisfying Assumption 1 where file popularity follows the Zipf distribution with parameter $\beta > 1$. Let $\widetilde{R}^*_{z_{>1}}$ denote the optimal transmission rate for any uncoded storage/service policy.

– If $\gamma \leq \dfrac{1}{\beta - 1}$,

$$\mathbb{E}\big[\widetilde{R}^*_{z_{>1}}\big] = \begin{cases} \Omega\left(m^{1-\mu(\beta-1)}\right) & \text{if } M \leq (1-\epsilon)n,\ \epsilon > 0, \\ \Omega\left(m^{\frac{2-\mu\beta}{\beta}}\right) & \text{if } M = n, \end{cases}$$

$$\mathbb{E}\big[\widetilde{R}^*_{z_{>1}}\big] \geq 0 \text{ if } M \geq (1+\epsilon)\, n,\ \epsilon > 0.$$

– If $\gamma > \dfrac{1}{\beta - 1}$,

$$\mathbb{E}\big[\widetilde{R}^*_{z_{>1}}\big] = \Omega\left(m^{1-\mu(\beta-1)}\right) \text{ if } M = \mathrm{o}\left(m^{\frac{1}{\beta-1}}\right),$$

$$\mathbb{E}\big[\widetilde{R}^*_{z_{>1}}\big] \geq 0 \text{ if } M = \Omega\left(m^{\frac{1}{\beta-1}}\right).$$

Remark 1: Note that this result only depends on the cumulative cache memory and is valid for all storage profiles with the same amount of cumulative cache memory. This result is a generalization of a result in [4] which holds only if the number of files scales linearly with the number of caches, i.e., $\gamma = 1$.

C. Knapasack Storage + Match Least Popular Policy

Next, we analyze the performance of a policy called Knapsack Storage + Match Least Popular policy (KS+MLP), proposed in [4]. In [4], it was shown that the KS+MLP policy is orderwise optimal for the homogeneous setting if the number of caches scales linearly with the number of files. We first make suitable modifications to the policy to incorporate heterogeneity in memory across caches and analyze its performance for more general storage profiles. We describe the modified KS+MLP policy in detail for the sake of completeness.

The KS+MLP policy comprises of two phases: the *placement phase* and the *delivery phase*.

1) Placement Phase: In the *placement phase*, the goal is to determine what to store on each cache. This task is completed in two steps.

Knapsack Storage: Part 1 – In this part, we decide how many caches store copies of each content by solving a Fractional Knapsack problem. In the Fractional Knapsack problem, each object has two attributes, namely, a weight and a value, and the knapsack has a finite weight capacity. The goal is to determine which objects should be added to the knapsack to maximize their cumulative value while the weight constraint of the knapsack is not violated. In the KS+MLP policy, each file corresponds to an object. The weight of an object/file corresponds to the number of caches on which it will be replicated if selected. The weights are chosen such that with high probability, all requests for that file can be served using the caches. More specifically, if file popularity follows the Zipf distribution with parameter $\beta > 1$, the weight of File i, denoted by w_i is assigned the following values.

$$w_i = \begin{cases} m, & \text{if } i = 1 \\ \left\lceil \left(1 + \frac{p_1}{2}\right) \widetilde{m} p_i \right\rceil, & \text{if } 1 < i \leq n_1, \\ \left\lceil 4 p_1 (\log m)^2 \right\rceil, & \text{if } n_1 < i \leq n_2, \\ \left\lceil \frac{1}{\delta} + 1 \right\rceil, & \text{if } n_2 < i \leq n, \end{cases} \tag{1}$$

where $n_1 = \frac{(\widetilde{m} p_1)^{\frac{1}{\beta}}}{(\log m)^{\frac{2}{\beta}}}$, and $n_2 = m^{\frac{1+\delta}{\beta}}$ for some $\delta > 0$. The value of File i is the probability that it is requested at least once in a batch of requests. The weight capacity of the cache system is equal to the cumulative cache memory. Using these parameter values, we solve the following Fractional Knapsack problem:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} x_i (1 - (1-p_i)^{\widetilde{m}}) \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i w_i \leq M, \\ & 0 \leq x_i \leq 1,\ \forall i. \end{aligned}$$

If the solution to the above Knapsack problem gives $x_i = 1$, we store w_i copies of File i else we don't store File i.

Knapsack Storage: Part 2 – The previous step determines how many copies of each file will be stored on the caches. The next task is to store the copies of files on caches. File copies

are sorted in increasing order of the corresponding file index. For example, consider a system consisting of five caches with $k_1 = 3, k_2 = 2, k_3 = 2, k_4 = 1, k_5 = 1$ units of memory. Say the solution for Knapsack Storage: Part 1 gives $x_1 = x_2 = x_3 = x_4 = x_5 = 1$ and 0 otherwise, and $w_1 = 4$, $w_2 = 2$, $w_3 = w_4 = w_5 = 1$. The sorted list of file copies is illustrated in Figure 3. Recall that caches are indexed in decreasing order of memory. The sorted list of file copies is stored on the caches in a round robin manner, i.e., the next file copy is placed on the next cache which has a memory slot available, see Figure 3 for an illustration.

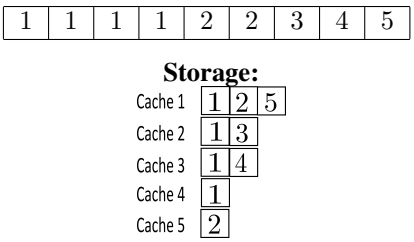

Fig. 3. Illustration of Knapsack Storage: Part 2 for a system with five caches.

2) Delivery Phase: In the *delivery phase*, requests are allocated to caches for service using the Match Least Popular policy (MLP), such that each cache is matched to at most one request. All the matched requests are served by the corresponding caches and all the unmatched requests are assigned to the central server. As the name suggests, the Match Least Popular policy matches requests for unpopular files before matching requests for popular files to caches. Refer to Figure 4 for a formal definition.

1: initialize $i = n$, set of idle caches $= \{1, 2, ..., m\}$.
2: **if** the number of requests for File i is more than the number of idle caches storing File i, **then**
3: goto Step 8.
4: **else**
5: match requests for File i to idle caches storing File i, chosen uniformly at random.
6: update the set of idle caches.
7: **end if**
8: $i = i - 1$, goto Step 2.

Fig. 4. Match Least Popular – *Matches requests to caches.*

The next theorem evaluates the performance of the KS+MLP policy for a particular sub-class of heterogeneous distributed cache systems.

Theorem 4: Consider a distributed cache system satisfying Assumption 1 where file popularity follows the Zipf distribution with parameter $\beta > 1$ and the top (largest) $\Omega(m^{2-\beta+\delta})$ caches, for any $\delta > 0$ have the same storage size. We have no restrictions on the storage sizes of the other (smaller) caches. Let $\mathbb{E}\big[\widetilde{R}^{\text{KS}}_{z_{>1}}\big]$ denote the expected transmission rate of the KS+MLP policy for this system.

– If $\gamma \leq \dfrac{1}{\beta - 1}$

$$\mathbb{E}\big[\widetilde{R}^{\text{KS}}_{z_{>1}}\big] = \begin{cases} \text{O}\left(m^{1-\mu(\beta-1)}\right) & \text{if } M \leq (1-\epsilon)n,\ 0 < \epsilon < 1, \\ \text{O}\left(m^{\frac{2-\mu\beta}{\beta}}\right) & \text{if } M = n \\ \text{O}\left(1\right) & \text{if } M \geq (1+\epsilon)\, n,\ \epsilon > 0. \end{cases}$$

– If $\gamma > \dfrac{1}{\beta - 1}$

$$\mathbb{E}\big[\widetilde{R}^{\text{KS}}_{z_{>1}}\big] = \begin{cases} \text{O}\left(m^{1-\mu(\beta-1)}\right) & \text{if } M = \text{o}\left(m^{\frac{1}{\beta-1}}\right), \\ \text{O}\left(1\right) & \text{if } M = \Omega\left(m^{\frac{1}{\beta-1}}\right). \end{cases}$$

Remark 2: The key takeaways from this result are:

– If the top $\Omega(m^{2-\beta+\delta})$ caches, for any $\delta > 0$ have the same memory size, then KS+MLP results match orderwise with the lower bounds in Theorem 3. Hence, in this case, the KS+MLP policy is orderwise optimal.
– Homogeneous systems have all caches with equal memory, and thus Theorem 4 also holds for homogeneous systems.

Combining Theorems 3 and 4 we have the following result.

Corollary 3: Consider a distributed cache system satisfying Assumption 1 where file popularity follows the Zipf distribution with parameter $\beta > 1$. If the top $\Omega(m^{2-\beta+\delta})$ caches, for any $\delta > 0$ have the same memory size, the performances of the optimal schemes for heterogeneous and homogeneous systems are orderwise equal.

Compare the above result with Corollary 2 for $\beta < 1$, which considers a heterogeneous cache system that divides a cumulative cache memory of $M = \Theta(n \cdot m^{\delta}), \delta > 0$, amongst two classes of caches: '*rich*' caches with larger storage size and '*poor*' caches with smaller storage. Corollary 2 shows that even if only a constant fraction of the caches are restricted to be poor, it can cause significant disparity between the performances of heterogeneous and homogeneous systems. On the other hand, in the same setting for $\beta > 1$, Corollary 3 shows that even if as many as $m - \Omega(m^{2-\beta+\delta})$ caches are restricted to be poor with only one unit of memory, the performance of the system will be orderwise the same as the homogeneous system. This suggests that caching systems are more tolerant to heterogeneity in storage under Zipf distributions with parameter $\beta > 1$ than under Zipf distributions with parameter $\beta < 1$.

Intuitively, this difference can be explained as follows. When content popularity is lopsided ($\beta > 1$), under the KS+MLP policy, caches with limited storage are used to serve requests for popular contents and the caches with large storage which store a mixture of some popular and a large number of unpopular contents typically are allocated to serve requests for unpopular contents. This ensures that the low storage caches are also utilized, unlike the case when content popularity is comparable across files. Since most caches are utilized, the number of requests served by the central server is small. As a result, the effect of storage heterogeneity is lower for lopsided

content popularity distributions as compared to distributions where it is comparable across files.

Corollary 3 describes a sufficient condition under which the performances of the homogeneous and heterogeneous systems remain comparable. Our next result characterizes a degree of heterogeneity sufficient to ensure that the performance of the heterogeneous system is orderwise inferior to that of a homogeneous system with the same amount of cumulative cache memory.

Theorem 5: Consider a distributed cache system satisfying Assumption 1 where file popularity follows the Zipf distribution with parameter $\beta > 1$. Let $\widetilde{R}^*_{z>1}$ denote the optimal transmission rate for any uncoded storage/service policy. If $\exists$ a subset $\mathcal{S}$ of caches with cumulative memory $|M_{\mathcal{S}}|$ such that

- $|\mathcal{S}| \geq m - m^{1-\mu(\beta-1)-\delta}$, for any $\delta > 0$ and
- $|M_{\mathcal{S}}| \leq (1-\epsilon)n$, for any $\epsilon > 0$,

$$\text{then, } \mathbb{E}[\widetilde{R}^*_{z>1}] \geq \Omega\big(m^{1-\mu(\beta-1)}\big).$$

We thus conclude that if there is a large enough set of caches (with cardinality $m - m^{1-\mu(\beta-1)-\delta}$, for any $\delta > 0$) with cumulative storage less than a constant fraction of the catalog size, the expected transmission rate can't be made arbitrarily small, irrespective of the total cache memory in the system.

Example: Consider a heterogeneous distributed cache system with m caches and $n = cm$ $(c > 1)$ files with content popularity following the Zipf distribution with $\beta > 1$. We have two classes of caches: 'rich' and 'poor'. Let the total cumulative memory in the system be $M = (1+\epsilon)n = (1+\epsilon)cm$ for some $\epsilon > 0$. Let m_1 denote the number of rich caches, each of which has $k \gg 1$ units of memory. The remaining $m - m_1$ poor caches each have 1 unit of memory, see Figure 5 for an illustration. Thus, we have $m - m_1 + m_1 k = M = (1+\epsilon)n$. For some small $\delta > 0$, Figure 5 depicts two systems with the same total cumulative memory, in which the number of rich caches is $m_1 = m^{2-\beta-\delta}$ and $m_1 = m^{2-\beta+\delta}$ respectively. For the former system which has fewer number of rich caches, the expected rate grows as $\Omega(m^{2-\beta})$ from Theorem 5. On the other hand, for the latter system which has more rich caches, Corollary 3 shows that the KS+MLP policy achieves $\mathrm{o}(1)$ rate. So for some small δ, modifying the storage profile to change the number of rich caches from $m^{2-\beta-\delta}$ to $m^{2-\beta+\delta}$ can have a dramatic impact on the server transmission rate.

IV. Simulation Results

In Section III, we presented asymptotic results as the system size m grows, which compare the effects of storage heterogeneity on the server transmission rate as a function of β (or as a function of the popularity profile). In this section, we simulate finite size cache systems and empirically validate some of our theoretical findings.

First, we consider a system which consists of m caches with total memory M units, $n = m$ files with popularity following the Zipf distribution with $\beta = 0.3$, and $\widetilde{m} = 0.97m$ requests. Similar to the example in the previous section, we consider two classes of caches: 'rich' and 'poor', i.e., out of the m caches, m_1 caches (rich caches) each have k units of memory

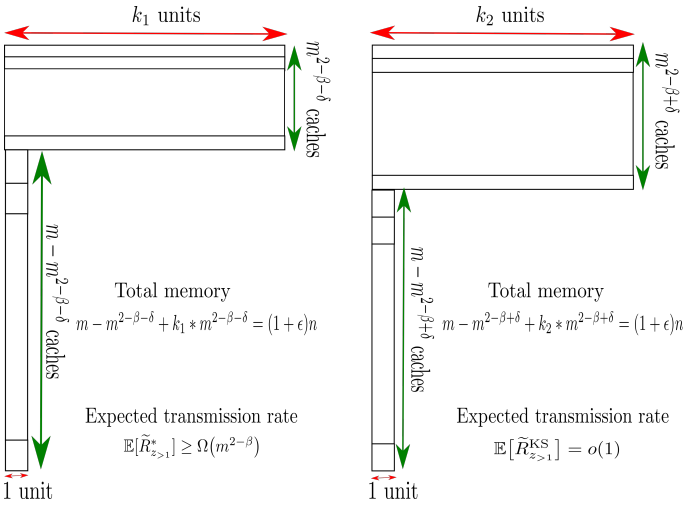

Fig. 5. Impact of storage heterogeneity on server transmission rates

and the remaining $m - m_1$ caches (poor caches) each have only 1 unit of memory. As the value of m_1 decreases, the memory is concentrated among fewer caches. For this system, we simulate the PPMM policy as described in Section III-A and consider the server transmission rate, averaged over 100 experiments.

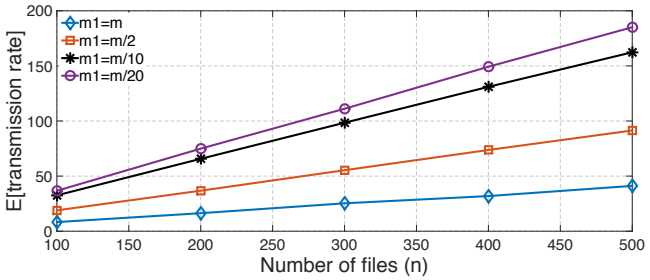

Fig. 6. Plot of the average transmission rate vs the number of files (n) for PPMM policy with $m_1 = \{m, \frac{m}{2}, \frac{m}{10}, \frac{m}{20}\}$, for a system where the number of caches (m) = n, the number of requests ($\widetilde{m}$) = $0.97n$, the Zipf parameter (β) = 0.3, and the total memory (M) = $3n$.

In Figure 6, we fix the total memory to $M = m_1 k + m - m_1 = 3m$ units and plot the average transmission rate as a function of number of files n for various values of m_1. As expected, *(i)* the transmission rate increases with n, and *(ii)* for any fixed value of n, the transmission rate increases drastically as the number of rich caches m_1 decreases. As our result in Corollary 2 suggests, there is significant difference between the homogeneous and heterogeneous cases.

In Figure 7, we fix $m = n = 400$ and plot the average server transmission rate as a function of the cache size k of each of the m_1 rich caches, for various values of m_1. As we increase k, we expect the transmission rate to decrease initially until all the rich caches serve one request each, and remain constant thereafter since the storage capacity of the poor caches is fixed throughout to 1 unit. As expected, *(i)* for the homogeneous case, the average transmission rate decreases exponentially with k until it reaches 0, and *(ii)* for the heterogeneous case,

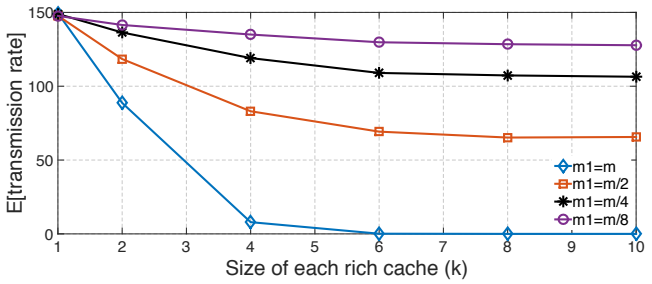

Fig. 7. Plot of the mean transmission rate vs cache size (k) of each of the m_1 rich caches for PPMM policy with $m_1 = \{m, \frac{m}{2}, \frac{m}{4}, \frac{m}{8}\}$, for a system where the number of caches (m) = the number of files (n) = 400, the number of requests ($\widetilde{m}$) = $0.97n$, and the Zipf parameter (β) = 0.3.

the average transmission rate decreases initially and remains constant after a certain k, depending upon the heterogeneity level (m_1).

Next, we consider a system which consists of m caches with total memory M units, $n = 5m$ files with popularity following the Zipf distribution with $\beta = 1.2$, and $\widetilde{m} = 0.97m$ requests. As before, we consider m_1 rich caches and $m - m_1$ poor caches. For this system, we simulate the KS+MLP policy as described in Section III-B and consider the server transmission rate, averaged over 100 iterations.

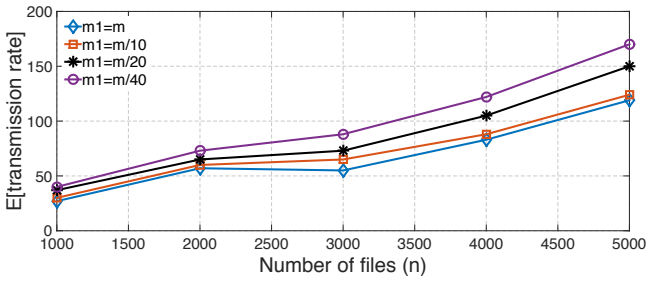

Fig. 8. Plot of the average transmission rate vs the number of files (n) for KS+MLP policy with $m_1 = \{m, \frac{m}{10}, \frac{m}{20}, \frac{m}{40}\}$, for a system where the number of caches (m) = $\frac{n}{5}$, the number of requests ($\widetilde{m}$) = $0.97n$, the Zipf parameter (β) = 1.2, and the total memory (M) = $3n$.

In Figure 8, we fix the total memory to $M = m_1 k + m - m_1 = 3m$ units and plot the average transmission rate as a function of number of files n for various values of m_1. As expected, *(i)* the transmission rate increases with n, and *(ii)* for any fixed value of n, unlike the $\beta = 0.3$ case (plotted in Figure 6), the change in transmission rate for different values of m_1 is small. This is in line with our result in Corollary 3, which suggests that the performances of the homogeneous and heterogeneous systems are similar.

V. PROOFS

In this section, we prove some of the results stated in Section III. Refer to [18] for the remaining proofs. [4] Note that we are interested in orderwise results. In the rest of this section, we will use c_i, where $i \in \mathbb{N}$, to represent positive constants.

[4]Proofs for homogeneous distributed caches systems are given in [4], [6].

A. Proof of Theorem 1

We analyze the performance of PPMM policy discussed in Section III for $\beta \in [0, 1)$ using ideas from the proof of Proposition 1 in [17] which looks at the setting where the request arrival process is Poisson and $\gamma = 1$. We first show that with high probability, there exists a fractional matching between the set of requests and the set of caches. By the total unimodularity of adjacency matrix, the existence of a fractional matching implies the existence of an integral matching [17]. Please refer to [18] for the details.

B. Proof of Theorem 2

We use the following lemmas to prove Theorem 2.

Lemma 1: For a Binomial random variable X with mean μ, for all $0 \leq \delta \leq 1$,

$$\mathbb{P}(X \leq (1+\delta)\mu) \leq e^{-\delta^2 \mu/3}.$$

Lemma 2: For a content delivery system satifying Assumption 1 with file popularity following the Zipf distribution with parameter $\beta \in [0, 1)$, let d_i represents the number of requests for File i in a batch. Then, for any $\delta > 0$,

$$\mathbb{P}\big(d_i \leq 2m^{\max\{0,1-\mu(1-\beta)\}+\delta}\big) = \mathrm{O}\Big(e^{-m^{\max\{0,1-\mu(1-\beta)\}+\delta}}\Big).$$

Proof: The popularity of File 1 is $p_1 = \frac{1}{\sum_{i=1}^{n} i^{-\beta}} \leq \frac{1}{n^{1-\beta}}$ for large n. Under Assumption 1, the number of requests for File 1 is Bin$(\widetilde{m}, p_1)$ and the expected number of requests is $\leq m^{1-\gamma(1-\beta)}$. Consider a new Binomial random variable X with mean $m^{\max\{0,1-\gamma(1-\beta)\}+\delta}$. By Lemma 1,

$$\begin{aligned}\mathbb{P}\big(d_i \leq 2m^{\max\{0,1-\gamma(1-\beta)\}+\delta}\big) &\\ \leq \mathbb{P}\big(X \leq 2m^{\max\{0,1-\gamma(1-\beta)\}+\delta}\big) &\\ = \mathrm{O}\Big(e^{-m^{\max\{0,1-\gamma(1-\beta)\}+\delta}}\Big).&\end{aligned}$$

■

We now prove Theorem 2 which evaluates the performance of a heterogeneous distributed cache system for $\beta \in [0, 1)$.

Proof: (Theorem 2 – *Lower bound on transmission rate*)
If a cache stores k units of data, the probability of the cache being idle is $\geq \big(1 - (\frac{k}{n})^{1-\beta}\big)^{\widetilde{m}}$. If $(\frac{k}{n})^{1-\beta} = \frac{1}{c_1 m}$, i.e., $k = \Theta(m^{\gamma - \frac{1}{1-\beta}})$, and $c_2 m$ caches have memory less than k units, then the expected number of idle caches is $\geq c_2 m e^{-\frac{\rho}{c_1}}$. Hence, the expected number of unserved requests is $\geq \widetilde{m} - m + c_2 m e^{-\frac{\rho}{c_1}}$. If $c_2 > \frac{1-\rho}{e^{-\frac{\rho}{c_1}}}$, then the number of unserved requests is $\Theta(m)$. From Lemma 2, no file is requested more than $m^{\max\{0,1-\gamma(1-\beta)\}+\delta}$ times for any $\delta > 0$. Hence, the expected transmission rate between the central server and the root-node is $\Omega\big(\frac{m}{m^{\max\{0,1-\gamma(1-\beta)\}+\delta}}\big) = \omega(1)$. ■

Proof: (Theorem 2 – *Performance of PPMM*)
Consider a new system (System B) by ignoring low memory ($k_i < m^{\gamma-\delta}$) caches, *i.e.*, System B contains $m' = (\rho + c_3)m$ caches, $n = \big(\frac{m'}{\rho + c_3}\big)^{\gamma}$ files, total memory is $M = m' * m^{\gamma-\delta}$, and receives $m' = \frac{\rho}{\rho+c_3} m'$ requests. Let $\mathbb{E}[R_B]$ is

expected transmission rate in System B. System B satisfies the conditions of Corollary 1. Therefore,

$$\mathbb{E}\big[\widetilde{R}^{\text{PPMM}}_{z_{[0,1)}}\big] \leq \mathbb{E}[R_B] = \text{o}(1).$$

■

C. Proof of Theorem 3

We use Proposition 1 in [4] to prove this theorem. The proof involves evaluating the lower bound characterized in Proposition 1 in [4] for a system which satisfies Assumption 1. Refer to [18] for the details.

D. Proof of Theorem 4

We use the following lemmas to prove Theorem 4. Please refer to [18] for the proofs of the lemmas.

The first lemma states that with high probability, all the requests for files stored by the Knapsack Storage policy are served by the caches.

Lemma 3: Let $\mathcal{R} = \{i : x_i = 1\}$, where x_i is the solution of the fraction knapsack problem solved in Knapsack Storage: Part 1. Let E_2 be the event that if the top (largest) $\omega(m^{2-\beta+\delta})$ caches, for any $\delta > 0$, have the same storage size, the Match Least Popular policy matches all requests for all contents in R to caches. Then, we have

$$\mathbb{P}(E_2) = 1 - \text{O}(ne^{-(\log m)^2}).$$

The next lemma evaluates the performance of the Knapsack Storage + Match Least Popular (KS+MLP) policy for the case where content popularity follows the Zipf distribution.

Lemma 4: Consider a distributed content delivery system satisfying Assumption 1, and the top (largest) $\omega(m^{2-\beta+\delta})$ caches, for any $\delta > 0$ have the same storage size. Let $R_{\text{KS+MLP}}$ be the transmission rate for the Knapsack Storage + Match Least Popular policy when content popularity follows the Zipf distribution with Zipf parameter $\beta > 1$. Then for m large enough, we have

$$\mathbb{E}[R_{\text{KS+MLP}}] \quad \leq \quad \sum_{i \notin R} 1 - \left(1 - \frac{p_1}{i^\beta}\right)^{\widetilde{m}} + \text{O}(mne^{-(\log m)^2}),$$

where $p_1 = \left(\sum_{i=1}^{n} i^{-\beta}\right)^{-1}$, $\mathcal{R} = \{i : x_i = 1\}$, such that x_i is the solution of the fraction knapsack problem solved in Knapsack Storage: Part 1.

Proof: (Theorem 4)

Case 1: $M \leq (1-\epsilon)n,\ 0 < \epsilon < 1$

From Lemma 3, if we store File i on w_i caches and employ the Knapsack Storage Policy: Part 2, all the requests for File i in a batch are served locally with high probability. Consider an alternative storage policy which starts storing files $2, 3, \ldots,$ each on w_i caches respectively until the cache memory is exhausted. This policy stores Files 2, 3, $\ldots$, $\frac{M-(1-\frac{p_1}{2})m}{\lceil \frac{1}{\delta}+1 \rceil}$. Let $\mathbb{E}[R]$ be the expected transmission rate for this policy. By the definition of fractional knapsack problem, $\mathbb{E}\big[\widetilde{R}^{KS}_{z_{>1}}\big] \leq \mathbb{E}[R]$. Therefore,

$$\mathbb{E}\big[\widetilde{R}^{KS}_{z_{>1}}\big] = \text{O}\Big(m^{1-\mu(\beta-1)}\Big).$$

Case 2: $M = n$ – Refer to [18] for the details.

Case 3: $M \geq (1+\epsilon)n,\ \epsilon > 0$ – Refer to [18] for the details. ■

E. Proof of Theorem 5

Proof: In our system, assume that the cumulative memory of $m - c_4 m^{1-\mu(\beta-1)-\delta}$ caches (say low memory caches) is $\leq (1-\epsilon)n$. Consider a new system with $m + c_4 m^{1-\mu(\beta-1)-\delta}$ caches, such that m caches are similar to our system and the remaining $c_4 m^{1-\mu(\beta-1)-\delta}$ caches (say new caches), have $\frac{n}{m}$ units of memory each. From Theorem 3 Case 1, new caches + low memory caches can serve at most $\widetilde{m} - m^{1-\mu(\beta-1)}$ requests, and the remaining $c_4 m^{1-\mu(\beta-1)-\delta}$ caches can serve at most $c_4 m^{1-\mu(\beta-1)-\delta}$ requests. Hence,

$$\mathbb{E}[\widetilde{R}^{*}_{z_{>1}}] \geq \Omega\big(m^{1-\mu(\beta-1)}\big).$$

■

REFERENCES

[1] YouTube: http://www.youtube.com.

[2] Netflix: www.netflix.com.

[3] Cisco Whitepaper: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html.

[4] S. Moharir and N. Karamchandani, "Content replication in large distributed caches," *arXiv preprint arXiv:1603.09153*, 2016.

[5] S. Borst, V. Gupt, and A. Walid, "Distributed caching algorithms for content distribution networks," in *IEEE INFOCOM*, 2010, pp. 1–9.

[6] K. S. Reddy, S. Moharir, and N. Karamchandani, "Resource pooling in large-scale content delivery systems," in *Communications (NCC), 2017 Twenty-third National Conference on*. IEEE, 2017, pp. 1–6.

[7] D. Rossi and G. Rossini, "On sizing ccn content stores by exploiting topological information," in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*. IEEE, 2012, pp. 280–285.

[8] M. A. Abd-Elmagid, O. Ercetin, and T. ElBatt, "Cache-aided heterogeneous networks: Coverage and delay analysis," *arXiv preprint arXiv:1701.06735*, 2017.

[9] M. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.

[10] S. Wang, W. Li, X. Tian, and H. Liu, "Coded caching with heterogenous cache sizes," *arXiv preprint arXiv:1504.01123*, 2015.

[11] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Centralized coded caching with heterogeneous cache sizes," in *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*. IEEE, 2017, pp. 1–6.

[12] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3212–3229, 2016.

[13] Y. Liu, F. Li, L. Guo, B. Shen, S. Chen, and Y. Lan, "Measurement and analysis of an internet streaming service to mobile devices," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2240–2250, 2013.

[14] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *IEEE INFOCOM*, 1999, pp. 126–134.

[15] H. Yu, D. Zheng, B. Zhao, and W. Zheng., "Understanding user behavior in large scale video-on-demand systems," in *EuroSys*, 2006.

[16] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2012, pp. 310–315.

[17] M. Leconte, M. Lelarge, and L. Massoulie, "Bipartite graph structures for efficient balancing of heterogeneous loads," in *ACM SIGMETRICS*, 2012, pp. 41–52.

[18] K. S. Reddy, S. Moharir, and N. Karamchandani, "Effects of storage heterogeneity in distributed cache systems," 2018, https://www.dropbox.com/s/qoz2q17sh2lmsmh/heterogeneous