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: ksvr1532@gmail.com, 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].

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 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 $n, m \to \infty$.

A. Storage Model

Cache $i$ has the capacity to store $k_i$ units of data. Let $M = \sum_{i=1}^n k_i$ 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 $\tilde{m} = \rho m$ (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 $\beta$ is a positive constant known as the Zipf parameter. Small values of $\beta$ imply that content popularity is comparable across contents while larger values of $\beta$ correspond to lopsided popularity distributions. Typical values of $\beta$ 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 denoted 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.

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.

The more general setting where each cache can serve up to $a \geq 1$ 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.
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 $\beta$.
- 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., $k_i \geq 1$ for all $i$.
- Requests are received in batches of $m = \rho n$, 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]. 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 $R_{2[0,1]}^{PPMM}$ be the central server’s transmission rate for the PPMM policy described above. Then, we have that,

$$
E[R_{2[0,1]}^{PPMM}] = \begin{cases} 
O(m(n)) & \text{if } M < (1-\epsilon)n, \ \epsilon > 0, \\
O(m^2 e^{-M/n}) & \text{if } M \geq n.
\end{cases}
$$

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 $R_{2[0,1]}^{PPMM}$ be the central server’s transmission rate for the PPMM policy described above. If $M \geq 3n \ln m = \Omega(n \ln m)$, then $E[R_{2[0,1]}^{PPMM}] = o(1)$.

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 distri-
(b) Performance of PPMM: Let $\bar{R}_{\alpha}^{PPMM}$ be the central
server’s transmission rate for the PPMM policy de-
scribed above. Then for any $c > 0$ and $\delta < 1$, if
a fraction of the $m$ caches, say $m_1 = (\rho + c)m$, have at least
$\Omega(n/m^\delta)$ units of memory, then,

$$E[\bar{R}_{\alpha}^{PPMM}] = o(1).$$

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 heteroge-
neous cache systems with the same cumulative memory.

**Corollary 2**: Let $\alpha \in (0, 1)$ and $\delta > 0$. Consider distributed caching systems satisfying
Assumption 1 where file popularity follows the Zipf distri-
bution with parameter $\beta \in [0, 1]$. Consider a heterogeneous
system such that $\alpha \cdot m$ caches each have at most $O(n/m^{1-\delta})$
units of memory and the remaining $(1-\alpha)m$ caches each
have memory $\Theta(n/m^\delta)$. Then,

$$E[\bar{R}_{\alpha}^{PPMM}] = o(1).$$

On the other hand, for a homogeneous system with the
same cumulative cache memory $M = \Theta(n/m^\delta)$ as the above
heterogeneous system, we have

$$E[\bar{R}_{\alpha}^{PPMM}] = o(1).$$

**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 distri-
bution with parameter $\beta > 1$. Let $\bar{R}_{z>1}$ denote the optimal
transmission rate for any uncoded storage/service policy.

- If $\gamma \leq \frac{1}{\beta - 1}$,

$$E[\bar{R}^{z>1}_*] = \Omega\left(\frac{m}{\mu^{\frac{1}{\beta - 1}}}\right)$$

- If $\gamma > \frac{1}{\beta - 1}$,

$$E[\bar{R}^{z>1}_*] = \Omega\left(\frac{m}{\mu}\right),$$

$$E[\bar{R}^{z>1}_*] = \Omega\left(\frac{m}{\mu^{\frac{1}{\beta}}}\right).$$

**Remark 1**: Note that this result only depends on the cumu-
lative 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. Knapsack 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 an-
alyze 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 place-
ment 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.

$$w_i = \begin{cases} 
  m, & \text{if } i = 1 \\
  \left[(1 + \frac{\mu}{2})\tilde{m}p_1\right], & \text{if } 1 < i \leq n_1, \\
  \left[4p_1(\log m)^2\right], & \text{if } n_1 < i \leq n_2, \\
  \left[\frac{m}{\mu} + 1\right], & \text{if } n_2 < i \leq n,
\end{cases}$$

(1)
For example, consider a system consisting of five caches with next task is to store the copies of files on caches. File copies the solution for Knapsack Storage: Part 1 gives $x = \frac{1}{\beta} - 1$. Assumption 1 where file popularity follows the Zipf distributed cache systems.

KS+MLP policy for this system.

- Homogeneous systems have all caches with equal memory sizes, for any $\delta > 0$ have the same storage size. We have no restrictions on the storage sizes of the other (smaller) caches.

Let $E[R_{x>1}]$ denote the expected transmission rate of the KS+MLP policy for this system.

- If $\gamma \leq \frac{1}{\beta - 1}$,
  
  $E[R_{x>1}^{KS}] = \begin{cases} 
  O\left(m^{1-\mu(\beta-1)}\right) & \text{if } M \leq (1-\epsilon)n \text{, } 0 < \epsilon < 1, \\
  O\left(m^{2-\mu}\right) & \text{if } M = n \\
  O\left(1\right) & \text{if } M \geq (1+\epsilon)n \text{, } \epsilon > 0.
  \end{cases}$

- If $\gamma > \frac{1}{\beta - 1}$,

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

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, 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 $\beta > 1$ 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.

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 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 otherwise 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 $R_{m,1}$ denote the optimal transmission rate for any uncoded storage/service policy. If $\exists \left| S \right|$ of caches with cumulative memory $\left| M_S \right|$ such that

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

then, $\mathbb{E}[R_{m,1}] \geq \Omega(m^{1-\mu(\beta-1)})$.

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 $m = m_1 + m_1k = M = (1 + \epsilon)n$. 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 $o(1)$ rate. So for some small $\delta$, 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.

![Fig. 5. Impact of storage heterogeneity on server transmission rates](image)

**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 $\beta$ (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 $\tilde{m} = 0.97m$ 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 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 $m_1 = \{m, \frac{m}{2}, \frac{m}{3}, \frac{m}{10}, \frac{m}{20}\}$, for a system where the number of caches ($m$) = $n$, the number of requests ($\tilde{m}$) = 0.97$n$, the Zipf parameter ($\beta$) = 0.3, and the total memory ($M$) = $3n$.](image)
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, 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 \(\bar{m} = 0.97m\) 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.

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 \(n\), 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.

V. PROOFS

In this section, we prove the results stated in Section III. We are interested in order wise results. In the rest of this section, we will use \(c_i\), where \(i \in \mathbb{N}\), to represent positive constants.

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 similar setting where the request arrival process is Poisson and \(\gamma = 1\).

Proof: (Theorem 1)

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

Trivial.

Case 2: \(M \geq n\)

Recall that each file is stored proportional to its popularity in PPMM policy. i.e., File \(i\) is stored on \(d_i = Mp_i\) caches. Let \(b_i\) denotes the number of requests for File \(i\) in a batch, \(b_i\) is Bin(\(\bar{m}, p_i\)), where, \(p_i = \frac{p_i}{\bar{m}}\). For \(b_i\) requests of File \(i\), we split each request into \(d_i\) sub-requests, each of size \(\frac{1}{d_i}\) units. Let \(d\) denote the set of files stored on Cache \(s\). For each \(i \in d\), we associate \(b_i\) sub-requests of File \(i\) to Cache \(s\). This allocation leads to a fractional matching where the total data served by each cache is less than 1 unit if \(\forall s \in \{1, 2, ..., m\}\),

\[
\sum_{i \in d} \frac{b_i}{d_i} \leq 1.
\]

By the total unimodularity of adjacency matrix, the existence of a fractional matching implies the existence of an integral matching [17].

For the Zipf distribution with parameter \(\beta \in [0, 1)\), we have that,

\[
p_n \geq \frac{1 - \beta}{n}.
\]

Let \(p^* = \frac{1 - \beta}{n}\). Hence, \(p_i \geq p^* \quad \forall i\).

\(^4\)Homogeneous basic versions of proofs are given in [4]. [6].
\[
\mathbb{P}\left( \sum_{i \in \partial s} \frac{b_i}{d_i} > 1 \right) \leq \inf_{s > 0} \frac{\mathbb{E}\left[ e^{\delta \sum_{i \in \partial s} \frac{b_i}{d_i}} \right]}{e^{\delta}}
\]
where
\[
\mathbb{E}\left[ e^{\delta \sum_{i \in \partial s} \frac{b_i}{d_i}} \right] \leq \prod_{i \in \partial s} \mathbb{E}\left[ e^{\frac{\delta b_i}{d_i}} \right]
\]
(due to negative associativity [18]).

Then, for all \( \delta > 0 \), by the Chernoff bound,
\[
\mathbb{P}\left( \sum_{i \in \partial s} \frac{b_i}{d_i} > 1 \right) \leq \inf_{s > 0} e^{-s} \prod_{i \in \partial s} \mathbb{E}\left[ e^{\frac{\delta b_i}{d_i}} \right]
\]
and the result follows.

Proof: The popularity of File 1 is \( p_1 = \frac{1}{\sum_{i=2}^{n} \frac{1}{i}} \leq \frac{1}{n^{3/2}} \) for large \( n \). Under Assumption [1] the number of requests for File 1 is Bin(\( m_1, p_1 \)) and the expected number of requests is \( m_1^{1-\gamma(1-\beta)} \). Consider a new binomial random variable \( X \) with mean \( m_1^{\max (0,1-\gamma(1-\beta)) + \delta} \). By Corollary 4
\[
\mathbb{P}(d_i \leq 2m_1^{\max (0,1-\gamma(1-\beta)) + \delta}) \leq \mathbb{P}(X \leq 2m_1^{\max (0,1-\gamma(1-\beta)) + \delta}) = o\left(e^{-m_1^{\max (0,1-\gamma(1-\beta)) + \delta}}\right).
\]

We now prove the Theorem 2, which evaluates the performance of heterogeneous system with \( \beta \in [0,1) \).

Proof: (Theorem 2 – Lower bound on transmission rate)
If a cache stores \( k \) units of data, the probability of the cache is being idle is \( \geq \left(1 - \left(\frac{k}{n}\right)^{1-\beta}\right)^{\tilde{m}} \). If \( \frac{k}{n}^{1-\beta} = \frac{1}{c_1 m} \) i.e., \( k = \Theta(m^{-1-\frac{1}{\beta}}) \), and \( c_2 m \) caches have memory less than \( k \) units, then the expected number of idle caches is \( \geq c_2 m e^{-\tilde{m}} \). Hence, the number of unserved requests is \( \geq \tilde{m} - m + c_2 m e^{-\tilde{m}} \). If \( c_2 > \frac{1}{e^{\frac{1}{\beta}}} \), 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 \( \Omega\left(\frac{m}{m^{\max (0,1-\gamma(1-\beta)) + \delta}}\right) = \omega(1) \).

Proof: (Theorem 2 – Performance of PPM)
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, \( n = \left(\frac{m'}{\rho + c_3}\right)^{\gamma} \) files, total memory is \( M = m' \ast m' \), and receives \( m' = \frac{1}{e^{\frac{1}{\beta}}} m' \) requests. Let \( \mathbb{E}[R_B] \) is expected transmission rate of System B. System B satisfies the conditions of Corollary 1. Therefore,
\[
\mathbb{E}[R_{PPMM}] \leq \mathbb{E}[R_B] = o(1).
\]

C. Proof of Theorem 4
We use the following lemma and proposition to prove Theorem 3 and Theorem 4.

Lemma 3: Let a content delivery system contains \( n = m^{\gamma} \) files, with file popularity follows the Zipf distribution with Zipf parameter \( \beta > 1 \). In a given time-slot, let \( d_i \) represents the number of requests for File \( i \). Let \( E_1 \) be the event that:

(a) \( d_i \geq 1 \) for \( i = O\left(m^{\frac{\beta}{3}}\right) \), where \( \epsilon > 0 \) is arbitrarily small constant,
(b) \( d_i \leq 2p_1 (\log m)^{3} \) for \( n_1 < i \leq n_2 \),
(c) \( d_i \leq \left(1 + \frac{4}{\beta} \right) mp_i \) for \( 1 \leq i \leq n_1 \),
where
\[
\frac{n_1}{c'} = \left(\frac{\tilde{m} p_i}{\log m}\right)^{\frac{1}{\beta}}, \quad \text{and} \quad n_2 = m^{\frac{1+\delta}{3}} \text{ for redsome } \delta > 0.
\]

Then we have
\[
\mathbb{P}(E_1) = 1 - O\left(n e^{-c' (\log m)^{2}}\right).
\]

Proof: We use the following lemma and proposition to prove Theorem 3 and Theorem 4.
Proof:
Follows from the Chernoff bound. \[4\] proves similar lemma (Lemma 2).

**Proposition 1:** Consider a distributed cache system satisfying Assumption \[1\] where file popularity follows the Zipf distribution with parameter $\beta > 1$. Let File $i$ size is $F_i$ bits, and $R^*$ denote the minimum transmission rate under the constraint that no stored bit can serve more than one user. Then, we have, the optimal solution has following structure:

\[
\mathbb{E}[R^*] \geq \sum_{i=1}^{n} \sum_{u=1}^{F_i} (1 - (1 - p_i)^{\tilde{m}}) - O^*,
\]

where, $O^* = \max \left( \max_{i=1}^{n} \sum_{u=1}^{F_i} x_{i,u} \left( 1 - (1 - p_i)^{\tilde{m}} \right) \right)$

s.t. $\sum_{i=1}^{n} \sum_{u=1}^{F_i} x_{i,u} \max\{\tilde{m}p_i, 1\} \leq M$,

$0 \leq x_{i,u} \leq 1, \forall i, u.$

**Proof:** \[4\] proves similar theorem (Theorem 1).

**Remark 3:** Proposition \[1\] gives a lower bound to our system. The lower bound depends on the solution to the fractional knapsack problem ($O^*$). In particular, if the content popularity follows the Zipf distribution with parameter $\beta$ (i.e., File $i$ is requested with popularity proportional to the $i^{-\beta}$), then the optimal solution has following structure: $\exists i_{\min}, i_{\max}$ such that

\[
x_{i,u} = \begin{cases} 1, & \text{if } i_{\min} < i < i_{\max}, \\ f_1, & \text{if } i = i_{\min}, \\ f_2, & \text{if } i = i_{\max}, \\ 0, & \text{otherwise}, \end{cases}
\]

where, $0 \leq f_1, f_2 \leq 1,$

\[
\sum_{i=1}^{n} b_i \sum_{u=1}^{F_i} x_{i,u} \max\{\tilde{m}p_i, 1\} \leq M.
\]

It follows that

\[
\mathbb{E}[R^*] = \Omega \left( \frac{\sum_{i=1}^{n} b_i}{1 - (1 - (1 - p_i)^{\tilde{m}})} \right).
\]

**Proof:** (Theorem \[3\])

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

Consider a new system with one cache of size $M$ units which can serve all the requests for the stored contents. It is clear that a lower bound on the transmission rate in the new system is also a lower bound on the transmission rate of the original system.

In the new system, we can store at most $M$ files. Therefore, all requests for the $n - M \geq \epsilon n$ files that are not stored have to be served by the central server. Therefore,

\[
\mathbb{E}[R^*_{z+1}] \geq \int_{M+1}^{n} \left( 1 - \left( 1 - \frac{p_1}{\beta} \right)^{\tilde{m}} \right) \, di
\]

\[
= \int_{M+1}^{n} \frac{\tilde{m}p_1}{\beta} (1 + o(1)) \, di
\]

\[
geq \frac{\tilde{m}p_1}{\beta - 1} \left( \frac{1}{(M+1)^{\beta-1}} - \frac{1}{n^{\beta-1}} \right) (1 + o(1))
\]

\[
= \Omega \left( \frac{m_1}{1 - \mu(\beta-1)} \right).
\]

**Case 2:** $M = n$

We use Proposition \[1\] to prove this result. Recall that the optimal solution to $O^*$ has the following structure: $\exists i_{\min} \geq 1$ and $i_{\max} \leq n$, such that,

\[
x_i = \begin{cases} 1, & \text{if } i_{\min} < i < i_{\max}, \\ f_1, & \text{if } i = i_{\min}, \\ f_2, & \text{if } i = i_{\max}, \\ 0, & \text{otherwise}, \end{cases}
\]

where, $0 \leq f_1, f_2 \leq 1.$

Let $i = \left[ \frac{\tilde{m}p_1}{\beta} \right].$ By the definition of fractional knapsack problem, we have that

\[
f_1 \tilde{m}p_1 \sum_{i=i_{\min}+1}^{i_{\max}} \tilde{m}p_i + \sum_{i=i_{\min}+1}^{i_{\max}} 1 + f_2 = M
\]

\[
\Rightarrow f_1 \tilde{m}p_1 \sum_{i=i_{\min}+1}^{i_{\max}} \tilde{m}p_i di + i_{\max} - 2 \tilde{i} + f_2 \leq M
\]

\[
\Rightarrow \sum_{i_{\min}+1}^{i_{\max}} \tilde{m}p_i di + i_{\max} \leq M + 3\tilde{i} - f_1 \tilde{m}p_1 \frac{1}{i_{\min}}.
\]

i.e.,

\[
\sum_{i_{\min}+1}^{i_{\max}} \tilde{m}p_i di + i_{\max} \leq M + 3\tilde{i} - f_1 \tilde{m}p_1 \frac{1}{i_{\min}}.
\]

In the new system, we can store at most $M$ files. Therefore, all requests for the $n - M \geq \epsilon n$ files that are not stored have to be served by the central server. Therefore,

\[
\mathbb{E}[R^*_{z+1}] \geq \int_{M+1}^{n} \left( 1 - \left( 1 - \frac{p_1}{\beta} \right)^{\tilde{m}} \right) \, di
\]

\[
= \int_{M+1}^{n} \frac{\tilde{m}p_1}{\beta} (1 + o(1)) \, di
\]

\[
geq \frac{\tilde{m}p_1}{\beta - 1} \left( \frac{1}{(M+1)^{\beta-1}} - \frac{1}{n^{\beta-1}} \right) (1 + o(1))
\]

\[
= \Omega \left( \frac{m_1}{1 - \mu(\beta-1)} \right).
\]
are requested at least once and are less popular than Content \(i\).

Hence, \(\forall\) caches. From Lemma 3 Part (a), all these files are requested \(i\) times.

Lemma 4 states that with high probability, all the requests for files stored by the Knapsack are served by the caches.

**Proof:** Before going into the proof we classify our caches into 2 types.

**Top caches:** which contain at least one file from the set \([2 : n_2]\).

**Bottom caches:** which does not contain files from \([2 : n_2]\).

if the Knapsack Storage policy decides to store Content \(i\), it stores it on \(w_i\) caches.

\[
\sum_{i=2}^{n_2} x_i w_i \leq \sum_{i=2}^{n_1} \left[ \left(1 + \frac{p_1}{2}\right) m \right] p_i + \sum_{i=n_1+1}^{n_2} \left[ 4 p_1 (\log m)^2 \right] 
\]

\[
\leq \left(1 + \frac{p_1}{2}\right) m \left(1 - p_1\right) 
+ (4 p_1 (\log m)^2 + 1) m^{\frac{1 + \delta}{\beta}} 
\leq (1 - c_7) m 
\]

for some \(c_7 > 0\).

Since the MLP policy matches requests starting from the least popular contents, we first focus on requests for contents less popular than Content \(n_2\). These less popular contents are stored \(\left\lceil \frac{1}{3} + 1 \right\rceil\) times.

**Bottom caches:**

Let \(m\) top (largest) caches have equal memory. Excluding File 1, the cumulative popularity of files of a cache belong to Bottom caches is

\[
\mathbb{P}_{bcp} \leq \frac{p_1}{(m^{1+\delta})^2} + \frac{p_1}{(m^{1+\delta} + m)^2} + \sum_{i=1}^{n} \frac{p_1}{(m^\delta + m + im^\delta)^2} 
\]

\[
\leq O\left(\frac{p_1}{m^{1+\delta} + m} + p_1 m^{1-f} + \frac{p_1}{m^\delta \sum_{i=m-1}^{n} \frac{1}{\psi^i}}\right) 
\]

\[
= \frac{p_1}{m^{1+\delta}} + \frac{p_1 m^{1-f}}{m^\delta} + \frac{p_1}{m^{1+\delta} \sum_{i=m-f}^{n} \frac{1}{\psi^i}} 
\]

\[
\leq \frac{1}{m^{(1-f)\beta-1} - \frac{1}{m^{\beta-1}}} 
\]

If \(F \geq 2 - \beta + \delta\),

\[
\mathbb{P}_{bcp} = O\left(\frac{1}{m^{1+\delta}}\right). 
\]
Let $X_{b,i}$ denote the number of requests arrived for files in the Cache $i$ except File 1. Then $\mathbb{E}[X_{b,i}] = O\left(\frac{1}{m^\beta}\right)$. By the Chernoff bound in Lemma 1, $\mathbb{P}(X_{b,i} \geq 1) = O\left(\frac{1}{m^\beta}\right)$. For $j \in [n_2, n]$, File $j$ is stored on $\lceil \frac{j}{n-1} \rceil$ caches. Hence, by the union bound, the probability that File $j$ unmatched is $O(m^{-\delta})$.

**Top caches:**

Excluding files belong to $[1 : n_2]$, the cumulative popularity of files of a cache belong to Top caches is

$$\mathbb{P}_{\text{tcp}} \leq \frac{p_1}{(c_3 m)^\beta} + \sum_{i=1}^n \frac{p_1}{(c_3 m + i m^2)^\beta} = c_8 \geq c_7$$

$$= O\left(\sum_{i=0}^{m^{-\beta}} \frac{p_1}{(c_3 m)^\beta} + \sum_{i=m^{-1}}^n \frac{p_1}{(m + i m^2)^\beta}\right) = O\left(\frac{p_1 m^{-1}}{(c_3 m)^\beta} + \frac{p_1}{m^\beta} \sum_{i=m^{-1}}^n \frac{1}{i^{\beta}}\right)$$

$$= \frac{p_1 m^{-1}}{(c_3 m)^\beta} + \frac{p_1}{(\beta - 1) m^\beta} \left[\frac{1}{(1-1/\beta-1)} - \frac{1}{n^{\beta-1}}\right]$$

If $\beta \geq 2 - \beta + \delta$, $\mathbb{P}_{\text{tcp}} = O\left(\frac{1}{m^{1+\delta}}\right)$.

Let $X_{t_1,i}$ denote the number of requests arrived for files in the Cache $i$ except for files ranked between 1 and $n_2$. Then $\mathbb{E}[X_{t_1,i}] = O\left(\frac{1}{m^\beta}\right)$. By the Chernoff bound in Lemma 1, $\mathbb{P}(X_{t_1,i} \geq 1) = O\left(\frac{1}{m^\beta}\right)$. For $j \in [n_2, n]$, File $j$ is stored on $\lceil \frac{j}{n-1} \rceil$ caches. Hence, by the union bound, the probability that File $j$ such that $j \in [n_2, n]$ unmatched is $O(m^{-\delta})$.

Next, we focus on files ranked between 2 and $n_2$. Let $D_i$ be the set of caches storing File $i$ for $2 \leq i \leq n_2$. From Equation 4 if files are stored according to Knapsack Storage: Part 2, each cache stores at most one file with index $i$ such that $2 \leq i \leq n_2$.

We now focus on requests for contents ranked between $n_1$ and $n_2$. Each file is stored on $|D_i| = \lceil 4 p_1 (\log m^2) \rceil$ caches. Let $X_{b_2,i}$ be the number of requests for these caches from files ranked greater than $n_2$. Then $\mathbb{E}[X_{b_2,i}] = O\left(\frac{4 p_1 (\log m^2)}{m^\beta}\right)$. By the Chernoff bound in Lemma 1, $\mathbb{P}(X_{b_2,i} \geq 2 p_1 (\log m^2)) = O\left(\frac{1}{m^\beta} 2^{\beta (\log m^2)^2}\right)$. From Lemma 3 each content is requested not more than $2 p_1 (\log m^2)$ times w.h.p. Hence, by the union bound, with probability $\geq 1 - O\left(\frac{m (\log m^2)^2}{m^\beta}\right)$, all requests for contents in $R$ ranked between $n_1$ and $n_2$ are matched to caches by Match Least Popular.

We now focus on requests for contents ranked between 2 and $n_1$. File $i$ is stored on $|D_i| = \lceil (1 + \frac{p_1}{2}) m p_i \rceil$ caches. Let $X_{b_3,i}$ be the number of requests for these caches from files ranked greater than $n_2$. Then $\mathbb{E}[X_{b_3,i}] = O\left(\frac{m (\log m^2)}{m^\beta}\right)$. By the Chernoff bound in Lemma 1, $\mathbb{P}(X_{b_3,i} \geq \frac{p_1}{2} m p_i) = O\left(\frac{1}{m^\beta} (\log m^2)^2\right)$. From Lemma 3 each content is requested not more than $(1 + \frac{p_1}{2}) m p_i$ times w.h.p. Hence, by the union bound, with probability $\geq 1 - O\left(\frac{m (\log m^2)^2}{m^\beta}\right)$, all requests for contents in $R$ ranked between 2 and $n_1$ are matched to caches by Match Least Popular.

We now focus on the requests for Content 1. Recall that if the Knapsack Store + Match Least Popular (KS+MLP) policy for the case where the content popularity follows the Zipf distribution.

**Lemma 5:** Consider a distributed content delivery system satisfying Assumption [4] and the top (largest) $\omega(m^{1-\beta/4})$ caches, for any $\delta > 0$ have the same storage size. Let $R_{\text{KS+MLP}}$ be the transmission rate for the Knapsack Store + Match Least Popular policy when content popularity follows the Zipf distribution with Zipf parameter $\beta > 1$. Then for $m$ large enough, we have

$$\mathbb{E}[R_{\text{KS+MLP}}] \leq \sum_{i \in R} \left(1 - \frac{p_1}{m^\beta}\right) \vec{n}_i + O(m n e^{-\log m^2}),$$

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:** From Lemma 4, we know that, for $m$ large enough, with probability $\geq 1 - O\left(n e^{-\log m^2}\right)$, all requests for the contents stored in caches by the KS+MLP policy are matched to caches. Let $\vec{n}_i$ be the number of contents not in $\mathcal{R}$ (i.e., not cached by the KS+MLP policy) that are requested at least once in a given time-slot. Therefore,

$$\mathbb{E}[\vec{n}_i] = \sum_{i \in \mathcal{R}} \left(1 - p_1\right) \vec{n}_i,$$

and

$$\mathbb{E}[R_{\text{KS+MLP}}] \leq \mathbb{E}[\vec{n}] P(E_2) + m(1 - P(E_2)) \leq \mathbb{E}[\vec{n}] + O(m n e^{-\log m^2}).$$

**Proof:** (Theorem 4)

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

From Lemma 4 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, . . . , each on $w_i$ caches respectively until the cache memory is exhausted. This policy stores Files 2, 3, . . . , $\min\{M - \frac{1}{1+1}, m\}$. Let $\mathbb{E}[R]$ be the expected transmission rate for this policy. By the
definition of fractional knapsack problem, \( \mathbb{E}[\widehat{R}_{z+1}^{KS}] \leq \mathbb{E}[R] \).

Therefore,
\[
\mathbb{E}[\widehat{R}_{z+1}^{KS}] \leq 1 + \int_{m-(1-\frac{p}{\beta})m}^{m} (1 - p_i) \, dm = 1 + \int_{m-(1-\frac{p}{\beta})m}^{m} \tilde{m} p_i (1 + o(1)) \, dm
\]
\[
\leq 1 + \frac{2\tilde{m} p_i}{\beta - 1} \left[ \frac{1}{\left(M-(1-\frac{2}{\beta})m\right)^{\beta-1}} - \frac{1}{(n)^{\beta-1}} \right]
\approx O\left(m^{1-\mu(\beta-1)}\right)
\]

**Case 2: \( M = n \)**

Let Knapsack solution decides to store files from \( i_{\min} + 1 \) to \( i_{\max} \).

\[
\int_{i_{\min}}^{i_{\max}} \left[ \left(1 + \frac{p_1}{2}\right) \tilde{m} p_i + 4p_1 (\log m)^2 \right] n^{1+\beta} + \sum_{i_{\min}}^{i_{\max}} \approx M
\]

\[
\int_{i_{\min}}^{i_{\max}} \left(1 + \frac{p_1}{2}\right) \tilde{m} p_i + 4p_1 (\log m)^2 n^{1+\beta} + \sum_{i_{\min}}^{i_{\max}} \approx M
\]

\[
i_{\max} \geq M - 4p_1 (\log m)^2 n^{1+\beta} + \left(1 + \frac{p_1}{2}\right) m p_1 \left(\frac{m^{1-\beta}}{(\log m)^{\beta}} \right) - \left(\frac{1}{n^{1-\beta}}\right)
\]

Let \( i_{\min} = m^\alpha \) such that \( \alpha < \frac{1}{\beta} \) and substitute it in the above equation, we get

\[
i_{\max} = M(1-o(1)) \quad \text{(5)}
\]

\[
i_{\max} = \frac{1}{n} - c_9 m^{-\alpha(\beta-1)(1-o(1))} \quad \text{(6)}
\]

\[
\mathbb{E}[\widehat{R}_{z+1}^{KS}] \leq m^\alpha + \int_{i_{\max}}^{n} (1 - p_i) \tilde{m} di
\]

\[
= m^\alpha + \int_{i_{\max}}^{n} \tilde{m} p_i (1 + o(1)) di
\]

\[
= m^\alpha + \frac{2\tilde{m} p_i}{\beta - 1} \left[ \frac{1}{i_{\max}^{\beta-1}} - \frac{1}{n^{\beta-1}} \right]
\]

\[
= m^\alpha + \frac{\tilde{m} p_i}{\beta - 1} \left[1 - \left(\frac{i_{\max}}{n}\right)^{\beta-1}\right]
\]

From (5) and (6),

\[
= O\left(m^\alpha + O\left(m^{1-\mu(\beta-1)-\alpha(\beta-1)}\right)\right)
\]

Optimize over \( \alpha \), we will get

\[
\mathbb{E}[\widehat{R}_{z+1}^{KS}] = O\left(n^{1-\mu(\beta-1)}\right)
\]

**Case 3: \( M \geq (1 + \epsilon)n \), \( \epsilon > 0 \)**

If we store from Content \( t \) to Content \( n \), the total memory required is lesser than

\[
\int_{t}^{m \frac{\beta}{(\log m)^{\beta}}} \left(1 + \frac{p_1}{2}\right) \tilde{m} p_i + 4p_1 (\log m)^2 n^{1+\beta} + n
\]

\[
\leq \left(1 + \frac{p_1}{2}\right) m p_1 \left(\frac{m^{1-\beta}}{(\log m)^{\beta}} \right) - \left(\frac{1}{n} \right)
\]

\[
+ 4p_1 (\log m)^2 n^{1+\beta} + n. \quad (7)
\]

Hence, \( \exists t \), a constant such that, equation (7) is less than \( M \).

Hence,

\[
\mathbb{E}[\widehat{R}_{z+1}^{KS}] \leq t = \Theta(1).
\]

**E. Proof of Theorem 5**

**Proof:** In our system, assume that the cumulative memory of \( m - c_{10} m^{1-\mu(\beta-1)-\delta} \) caches (say low memory caches) is \( \leq (1-\epsilon)n \). Consider a new system with \( m + c_{10} m^{1-\mu(\beta-1)-\delta} \) caches, such that \( m \) caches are similar to our system and the remaining \( c_{10} m^{1-\mu(\beta-1)-\delta} \) 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 \( \tilde{m} - m^{1-\mu(\beta-1)} \) requests, and the remaining \( c_{10} m^{1-\mu(\beta-1)-\delta} \) caches can serve \( c_{10} m^{1-\mu(\beta-1)-\delta} \) requests. Hence,

\[
\mathbb{E}[\hat{R}_{z+1}^{*}] \geq \Omega\left(m^{1-\mu(\beta-1)}\right).
\]

**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. Gupta, 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 (ICC), 2017 Twenty-third National Conference on. IEEE, 2017, pp. 1–6.

[7] D. Rossi and G. Rossiini, “On sizing cn content stores by exploiting topological information,” in Computer Communications Workshops (INFCOM 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 heterogeneous 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. Sibi, “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] D. Dubhashi and D. Ranjan, “Balls and bins: A study in negative dependence,” BRICS Report Series, vol. 3, no. 25, 1996.