Proximity-Aware Balanced Allocations in Cache Networks

Ali Pourmiri*‡, Mahdi Jafari Siavoshani†, Seyed Pooya Shariatpanahi‡

*Department of Computer Engineering, University of Isfahan, Isfahan, Iran
†Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
‡School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran
{alipourmiri@gmail.com, mjafari@sharif.edu, pooya@ipm.ir}

***Abstract*—We consider load balancing in a network of caching servers delivering contents to end users. Randomized load balancing via the so-called *power of two choices* is a well-known approach in parallel and distributed systems that reduces network imbalance. In this paper, we propose a randomized load balancing scheme which simultaneously considers cache size limitation and proximity in the server redirection process.**

Since the memory limitation and the proximity constraint cause correlation in the server selection process, we may not benefit from the power of two choices in general. However, we prove that in certain regimes, in terms of memory limitation and proximity constraint, our scheme results in the maximum load of order $\Theta(\log\log n)$ (here n is the number of servers and requests), and at the same time, leads to a low communication cost. This is an exponential improvement in the maximum load compared to the scheme which assigns each request to the nearest available replica. Finally, we investigate our scheme performance by extensive simulations.

Keywords

Randomized Algorithms, Distributed Caching Servers, Request Routing, Load Balancing, Communication Cost, Balls-into-Bins, Content Delivery Networks.

I. Introduction

A. Problem Motivation

Advancement of technology leads to the spread of smart multimedia-friendly communication devices to the masses which causes a rapid growth of demands for data communication [1]. Although Telcos have been spending hugely on telecommunication infrastructures, they cannot keep up with this data demand explosion. Caching predictable data in network off-peak hours, near end users, has been proposed as a promising solution to this challenge. This approach has been used extensively in content delivery networks (CDNs) such as Akamai, Azure, Amazon CloudFront, etc. [2], [3], and mobile video delivery [4]. In this approach, a *cache network* is usually referred to as a set of caching servers that are connected over a network, giving content delivery service to end users.

In cache networks, load balancing is one of the most important challenges when assigning requests to servers. This assignment strategy is implemented either at network-side or client-side. In the first approach there is a centralized authority which maps requests to servers, while balancing out the load. This authority employs network status information to optimally allocate requests to servers, which often involves complex algorithms. However, in the latter, the clients choose their favorite servers in a distributed fashion. In this paper we focus on the distributed server selection approach.

Randomized load balancing via the so-called "power of two choices" is a well-investigated paradigm in parallel and distributed settings [5], [6], [7], [8]. In this approach, upon arrival of a request, the corresponding user will query about current load of two independently at random chosen servers, and then allocates the request to the least loaded server. Berenbrink et al. [9] showed that in this scheme after allocating m balls (requests, tasks, ...) to n bins (servers, machines, ...) the maximum number of balls assigned to any bin, called *maximum load*, is at most $m/n+O(\log\log n)$ with high probability. This only deviates $O(\log\log n)$ from the average load and the deviation depends on the number of servers. However, in many settings, selecting any two random servers might be infeasible or costly. For example *proximity principle* in CDNs for server selection is essential to reduce communication cost; i.e., each request should be redirected to a nearby server.

Considering this constraint, Kenthapadi and Panigrahi [10] proposed a model where n bins are organized as a d-regular graph. Corresponding to each ball, a node is chosen uniformly at random as the first candidate. Then, one of its neighbours is chosen uniformly at random as the second candidate and the ball is allocated to the one with the minimum load. Under this assumption, they proved that if the graph is sufficiently dense (i.e., the average degree is $n^{\Omega(\log\log n/\log n)}$), then after allocating n balls the maximum load is $\Theta(\log\log n)$ with high probability. Although the model fairly considers the proximity principle, due to cache limitations it cannot be directly applied in cache networks.

In summary, the proximity principle can be in tension with load balancing in many situations, as nearby users may be congested. This leads to a fundamental trade-off between the maximum load and the communication cost. Hence, designing a distributed assignment strategy to handle this trade-off optimally is a central and challenging goal in cache networks.

This research was in part supported by a grant from IPM.

1530-2075/17 $31.00 © 2017 IEEE
DOI 10.1109/IPDPS.2017.24

B. Problem Setting and Our Contributions

While many authors have used the idea of power of two choices in server-selection algorithms, theoretical foundations of this phenomena in the context of cache networks with communication cost, has not yet been investigated. In this paper, we consider a general cache network model that entails basic characteristics of many practical scenarios. We consider a grid network of n servers, each equipped with a cache of size M. Also there are n sequential file requests, from a library of size K, distributed among servers uniformly at random. Let us assume a popularity distribution $\mathcal{P} = \{p_1, \ldots, p_K\}$ for the library. We assume cache placement at each server is proportional to this popularity distribution. Every server either serves its requests or redirects them (via an assignment scheme) to other nodes which have cached the files. We define the maximum load of an assignment scheme as the maximum number of allocations to any single server after assigning all requests. The communication cost is the average number of hops required to deliver requested file to its request origin.

In the simplest assignment scheme, each request arrived at every server should be dispatched to the nearest file replica. This scheme results in the minimum communication cost, while ignoring maximum load of servers. We show that, for every constant $0 < \alpha < 1/2$, if $K = n$, $M = n^{\alpha}$, and $\mathcal{P}$ is a uniform distribution, this scheme will result in the maximum load in the interval $[\Omega(\log n / \log\log n), O(\log n)]$ with high probability[2] (w.h.p.). Moreover, for every constant $0 < \epsilon < 1$, if $K = n^{1-\epsilon}$ and $M = \Theta(1)$, then the maximum load is $\Theta(\log n)$ w.h.p. We also investigate the communication cost occurred in this scheme for Uniform and Zipf popularity distributions. In particular, we derive the communication cost of $\Theta(\sqrt{K/M})$ for the Uniform distribution.

In contrast, we propose a new scheme which considers both maximum load and communication cost objectives simultaneously. For each request, this scheme chooses two random candidate servers that have cached the request while putting a constraint on their distance r to the requesting node (i.e., the proximity constraint). Due to cache size limitation and the proximity constraint, current results in the balanced allocation literature cannot be carried over to our setting. Basically, we show that here the two chosen servers will become correlated and this might diminish the power of two choices. Since this correlation arises from both memory limitation and proximity principle, the main challenge we address in this paper is characterizing the regimes where we can benefit from the power of two choices and at the same time have a low communication cost.

In particular, suppose $0 < \alpha, \beta < 1/2$ be two constants and let $K = n$, $M = n^{\alpha}$, $r = n^{\beta}$, and $\mathcal{P}$ be a Uniform distribution. Then, provided $\alpha + 2\beta \geq 1 + 2(\log\log n / \log n)$, the maximum load is $\Theta(\log\log n)$ w.h.p., and the communication cost is $\Theta(r)$. Therefore, we deduce that if we set $M = n^{\alpha}$, for some constant $0 < \alpha < 1/2$, then it is sufficient to have $\beta = \frac{1-\alpha}{2} + \log\log n / \log n$ and hence $r = n^{\frac{1-\alpha}{2}} \log n$. This means that the communication cost is only $\log n$ factor above the communication cost achieved by the nearest replica strategy, which is $\Theta(\sqrt{K/M}) = \Theta(n^{\frac{1-\alpha}{2}})$.

C. Related Work

Load balancing has been the focus of many papers on cache networks [11], [12], [13], among which distributed approaches have attracted a lot of attention (e.g., see [14], [7], and [15]). Randomized load balancing via the power of two choices, is a popular approach in this direction [6]. Chen et al. [16] consider the two choices selection process, where the second choice is the next neighbor of the first choice. In [17] Xia et al. use the length of common prefix (LCP)-based replication to arrive at a recursive balls and bins problem. In [16] and [17], the authors benefit from the metaphor of power of two choices to design algorithms for randomized load balancing. In contrast, in this paper we follow a theoretical approach to derive provable results for cache networks with limited memory.

In [18] the authors consider the supermarket model for performance evaluation of CDNs. Although the work [18] considers the memory limitation into account, it does not consider the proximity principle which is a central issue in our paper. Liu et al. [19] study the setting where the clients compare the servers in terms of hit-rate (for web applications), or bit-rate (for video applications) to choose their favourite ones. Their setup and objectives are different from those we consider here. Moreover, they have not considered the effect of their randomized load balancing scheme on communication cost.

Additionally, the trade-off between proximity and load balancing in request routing has been considered in some works such as [20], [21], and [22]. Although these works have mentioned this trade-off, non of them provides a rigorous analysis. To the best of our knowledge, our paper is the first work characterizing the above trade-off in an analytical framework.

From the theoretical viewpoint, in the standard balls and bins model, each ball (request) picks two bins (servers) independently and uniformly at random and it is then allocated to the one with lesser load [5]. However, memory limitation and proximity principle in cache networks makes the bins choices correlated which resembles the balls and bins model with *related choices* (e.g., see [23], [10], [24], and [25]). Our result also resides in this category, which is specific to cache networks with memory limitation and proximity constraint.

The organization of the paper is as follows. In Section II, we present our notation and problem setup. Then, in Section III the *nearest replica strategy*, as the baseline scheme, is presented and its maximum load and communication cost are investigated. In Section IV, we propose the *proximity-aware two choices strategy*, which at the same time considers proximity of requests and servers, and benefits from the power of two choices. In order to do this, we first present some examples to shed light on different aspects of the problem. Then, we propose our main results in two different regimes,

[2] With high probability refers to an event that happens with probability $1 - 1/n^c$, for some constant $c > 0$.

namely $M = n^{\alpha}$, for every constant $0 < \alpha < 1/2$, and $M = K$. In Section V performance of these two schemes are investigated via extensive simulations. Finally, our discussions and future directions are presented in Section VI.

II. Notation and Problem Setting

A. Notation

Throughout the paper, with high probability refers to an event that happens with probability $1-1/n^c$, for some constant $c > 0$. Let $G = (V, E)$ be a graph with vertex set V and edge set E where $e(G) := |E|$. For $u \in V$ let $d(u)$ denote the degree of u in G. For every pair of nodes $u, v \in V$, $d_G(u, v)$ denotes the length of a shortest path from u to v in G. The neighborhood of u at distance r is defined as

$$B_r(u) := \{v : d_G(u, v) \leq r \text{ and } v \in V(G)\}.$$

Finally, we use $\mathrm{Po}(\lambda)$ to denote for the Poisson distribution with parameter λ.

B. Problem Setting

We consider a cache network consisting of n caching servers (also called cache-enabled nodes) and edges connecting neighboring servers forming a $\sqrt{n} \times \sqrt{n}$ grid. Thus, direct communication is possible only between adjacent nodes, and other communications should be carried out in a multi-hop fashion.

Remark 1. *Throughout the paper for the sake of presentation clarity we may consider a torus with n. This helps to avoid boundary effects of grid and all the asymptotic results hold for the grid as well.*

Suppose that the cache network is responsible for handling a library of K files $\mathcal{W} = \{W_1, \ldots, W_K\}$, whereas the popularity profile follows a known distribution $\mathcal{P} = \{p_1, \ldots, p_K\}$.

The network operates in two phases, namely, *cache content placement* and *content delivery*. In the cache content placement phase each node caches $M \leq K$ files randomly from the library according to their popularity distribution $\mathcal{P} = \{p_1, \ldots, p_K\}$ with replacement, independent of other nodes. Also note that, throughout the paper we assume that $M \ll K$, unless otherwise stated.

Consider a time block during which n files are requested from the servers sequentially. The server of each request is chosen uniformly at random from n servers. Let D_i denote the number of requests (demands) arrived at server i. Then for large n we have $D_i \sim \mathrm{Po}(1)$ for all $1 \leq i \leq n$.

For library popularity profile $\mathcal{P}$, we consider two probability distributions, namely, Uniform and Zipf with parameter γ. In the Uniform distribution we have

$$p_i = \frac{1}{K}, \quad i = 1, \ldots, K,$$

which considers equal popularity for all the files. In Zipf distribution the request probability of the i-th popular file is inversely proportional to its rank as follows

$$p_i = \frac{1/i^{\gamma}}{\sum\limits_{j=1}^{K} 1/j^{\gamma}}, \quad i = 1, \ldots, K,$$

for a given parameter $\gamma > 0$, which has been confirmed to be the case in many practical applications [26], [27].

For any given cache content placement, an assignment strategy determines how each request is mapped to a server. Let T_i denote the number of requests assigned to server i at the end of mapping process.

Now, for each strategy we define the following metrics.

Definition 1 (Communication Cost and Maximum Load)**.**

- *The* communication cost *of a strategy is the average number of hops between the requesting node and the serving node, denoted by C.*
- *The* maximum load *of a strategy is the maximum number of requests assigned to a single node, denoted by $L = \max_{1 \leq i \leq n} T_i$.*

III. Nearest Replica Strategy

The simplest strategy for assigning requests to servers is to allocate each request to the nearest node that has cached the file. This strategy, formally defined below, leads to the minimum communication cost, while does not try to reduce maximum load.

Definition 2 (Strategy I: Nearest Replica Strategy)**.** *In this strategy each request is assigned to the nearest node –in the sense of the graph shortest path distance– which has cached the requested file. If there are multiple choices ties are broken randomly.*

Consider the set of nodes that have cached file W_j, say S_j. According to Strategy I, each demand from node u for file W_j will be served by $\arg\min_{v \in S_j} d_G(u, v)$. This induces a Voronoi Tessellation on the torus corresponding to file W_j which we denote by $\mathcal{V}_j$. Then, alternatively, we can define Strategy I as assigning each request of file W_j to the corresponding Voronoi cell center.

In order to analyze the maximum load imposed on each node, we should investigate the size of such Voronoi regions. The following Lemma is in this direction.

Lemma 1. *Under the Uniform popularity distribution, the maximum cell size (number of nodes inside each cell) of $\mathcal{V}_j$, $1 \leq j \leq K$, is at most $O(K \log n/M)$ w.h.p. In particular, every Voronoi cell centered at any node is contained in a sub-grid of size $r \times r$ with $r = O\left(\sqrt{K \log n/M}\right)$. Furthermore, if $K = n^{1-\epsilon}$, for some constant $0 < \epsilon < 1$, and $M = \Theta(1)$, then there exists a Voronoi cell of size $\Theta(K \log n/M)$ w.h.p.*

Proof. Refer to [28, Appendix B]. □

Now, we are ready to present our main results for this section which characterize the maximum load of Strategy I, in Theorems 1 and 2.

Theorem 1. *Suppose that $K = n^{1-\epsilon}$, for some constant $0 < \epsilon < 1$, and $M = \Theta(1)$. Then, under Uniform distribution $\mathcal{P}$, Strategy I achieves maximum load of $L = \Theta(\log n)$ w.h.p.*

Proof. Consider node u which has cached a set of distinct files, say S, with $|S| \leq M$. Applying Lemma 1 shows that all Voronoi cells centered at u corresponding to cached files at u are contained in a sub-grid of size at most $O(K \log n/M)$ w.h.p. Also in each round, every arbitrary node requests for a file in S with probability $|S|/nK \leq M/nK$, as each request randomly chooses its origin and type. Hence, by union bound, a node in the sub-grid may request for a file in S with probability at most $O(K \log n/M) \cdot (M/nK) = O(\log n/n)$. Since there are n requests, the expected number of requests imposed to node u is $O(\log n)$. Now using a Chernoff bound (e.g., see [28, Appendix A]) shows that w.h.p. u has to handle at most $O(\log n)$ requests.

On the other hand, to establish a lower bound on the maximum load we proceed as follows. Lemma 1 shows that there exits a Voronoi cell in $\mathcal{V}_j$, for some j, such that the center node should handle the requests of at least $\Theta(K \log n/M)$ nodes w.h.p. Also each node in the cell may request for file W_j with probability $1/nK$. So on average there are $\Theta(\log n/M)$ requests imposed on the cell center. Similarly, by a Chernoff bound, one can see that this node experiences the load $\Theta(\log n/M)$, which concludes the proof for constant M. □

Remark 2. *It should be noted that the same result of $\Theta(\log n)$ for the maximum load can also be proved for the Zipf distribution. That is because the content placement distribution is chosen proportional to the file popularity distribution $\mathcal{P}$, and consequently this result is insensitive to $\mathcal{P}$. However, the proof involves lengthy technical discussions which we omit in this paper.*

Theorem 2. *Suppose that $K = n$ and $M = n^{\alpha}$, for some $0 < \alpha < 1/2$. Then, under the Uniform distribution, the maximum load is in the interval $[\Omega(\log n/ \log\log n), O(\log n)]$ w.h.p.*

Proof. Refer to [28, Appendix B]. □

Next, we investigate the communication cost of Strategy I in the following theorem.

Theorem 3. *Under the Uniform popularity distribution, Strategy I achieves the communication cost $C = \Theta(\sqrt{K/M})$, for every $M \ll K$. Under Zipf popularity distribution with $M = \Theta(1)$, it achieves*

$$C = \begin{cases} \Theta\left(\sqrt{K/M}\right) & : \quad 0 < \gamma < 1, \\ \Theta\left(\sqrt{K/M} \log K\right) & : \quad \gamma = 1, \\ \Theta\left(K^{1-\gamma/2}/\sqrt{M}\right) & : \quad 1 < \gamma < 2, \\ \Theta\left(\log K/\sqrt{M}\right) & : \quad \gamma = 2, \\ \Theta\left(1/\sqrt{M}\right) & : \quad \gamma > 2. \end{cases} \tag{1}$$

Proof. Refer to [28, Appendix B]. □

Theorem 3 shows how non-uniform file popularity reduces communication cost. The skew in file popularity is determined by the parameter γ which will affect the communication cost. For example, for $\gamma < 1$ communication cost is similar to the Uniform distribution, while for $\gamma > 2$, it becomes independent of K.

Since in Strategy I we have assigned each request to the nearest replica, Theorem 3 characterizes the minimum communication cost one can achieve. However, Theorems 1 and 2 show a logarithmic growth for the maximum load as a function of network size n. This imbalance in the network load is because in Strategy I each request assignment does not consider the current load of servers. A natural question is whether, at each request allocation, one can use a very limited information of servers' current load in order to reduce the maximum load. Also one can ask how does this affect the communication cost.

IV. Proximity-Aware Two Choices Strategy

Strategy I introduced in the last section will result in the minimum communication cost, while, the maximum load for that strategy is of order $\Omega\left(\log n/ \log\log n\right)$. In this section we investigate an strategy which will result in an exponential decrease in the maximum load, i.e., reduces maximum load to $\Theta\left(\log\log n\right)$, formally defined as follows.

Definition 3 (Proximity-Aware Two Choices Strategy). *For each request born at an arbitrary node u consider two uniformly at random chosen nodes from $B_r(u)$, that have cached the requested file. Then, the request is assigned to the node with lesser load. Ties are broken randomly.*

For the sake of illustration, first, we consider some examples in the following.

Example 1 ($M = K$ and $r = \infty$[3]). *In this example each node can store all the library and there is no constraint on proximity. As mentioned in Section I, the number of files that should be handled by each node (i.e., D_i for $i = 1, \ldots, n$) will be a $\mathrm{Po}(1)$ random variable. In this case, according to Strategy II, two random nodes are chosen from all network nodes and the request is assigned to the node with lesser load.*

Therefore, in terms of maximum load, this problem is reduced to the standard power of two choices model in the balanced allocations literature [5]. In this model there are

[3] It should be noted that $r \geq \sqrt{n}$ (including $r = \infty$) is equivalent to $r = \sqrt{n}$. Thus in this paper we use $r = \sqrt{n}$ and $r = \infty$ alternatively.

n bins and n sequential balls which are randomly allocated to bins. In every round each ball picks two random bins uniformly, and it is then allocated to the bin with lesser load [5]. Then it is shown that the maximum load of network is $L = \max_i T_i = \log\log n(1 + o(1))$ w.h.p. [5], which is an exponential improvement compared to Strategy I.

However, in contrast to Example 1, in cache networks usually each node can store only a subset of files, and this makes the problem different from the standard balls and bins model, considered in [5]. Here, due to the memory constraint at each node, the choices are much more limited than the $M = K$ case. In other words here we have the case of *related choices*. In the related choices scenario, the event of choosing the second choice is correlated with the first choice; this correlation may annihilate the effect of power of two choices as demonstrated in Example 2.

Example 2 ($K = n$, $M = \Theta(1)$, and $r = \infty$)**.** *In this regime, there is a subset of the library, say S with $|S| = \Theta(n)$, whose files are replicated in $[1, M]$ number of places. On the other hand, each file type is requested $\mathrm{Po}(1)$ times and hence w.h.p. there will be a file in S which is requested $\Theta(\log n/ \log\log n)$ times (e.g., see [29]). Since each file in S is replicated at most M times, requests for the file are distributed among at most M nodes and thus the maximum load of the corresponding nodes will be at least $\Theta(\log n/ \log\log n)/M$. Hence, due to memory limitation we cannot benefit from the power of two choices.*

Although Example 2 shows that memory limitation can annihilate the power of two choices this is not always the case. Example 3 shows that even for $M = 1$ for some scenarios we can achieve $L = O(\log\log n)$.

Example 3 ($K = n^{1-\epsilon}$ for every constant $0 < \epsilon < 1$, $M = 1$, and $r = \infty$)**.** *For any popularity distribution $\mathcal{P}$ where $\sum_{j=1}^{K} (p_j n)^{-c} = o(1)$, Strategy II achieves maximum load $L = O(\log\log n)$ w.h.p. Also, notice that Uniform and Zipf distributions satisfy this requirement, whenever $\epsilon \in \left(\frac{\gamma-1}{\gamma}, 1\right)$ for $\gamma \geq 1$, where γ is Zipf parameter.*

Roughly speaking, when $M = 1$, we may partition the servers based on their cached file and hence we have K "disjoint" subsets of servers. Similarly there are K request types where each request should be addressed by the corresponding subset of servers. Thus, here we have K disjoint Balls and Bins sub-problems, and the sub-problem with maximum load determines the maximum load of the original setup. The reason that here, in contrast to Example 2, we can benefit from power of two choices is the assumption of $K \ll n$.

For a formal proof of above claim, refer to [28, Appendix C].

Above examples bring to attention the following question.

Question 1. *In view of the memory limitation at each server in cache networks, what are the regimes (in terms of problem parameters) one can benefit from the power of two choices to balance out the load?*

Addressing Question 1, for the general $M > 1$ case, is more challenging than Example 3 and needs a completely different approach. The simplicity of case $M = 1$ is that there is no interaction between K Balls and Bins sub-problems. On the other hand, consider $M > 1$. If a request, say W_j, should be allocated to a server then the load of two candidate bins that have cached W_j should be compared. However, load of other file types will also be accounted for in this comparison. So there is flow of load information between different sub-problems which makes them entangled.

In all above examples, we have not considered the proximity constraint, i.e., $r = \infty$, yet. This results in a fairly high communication cost $C = \Theta\left(\sqrt{n}\right)$. However, in general since parameter r controls the communication cost, it can be chosen to be much less than the network diameter, i.e., $\Theta(\sqrt{n})$. This proximity awareness introduces another source of correlation (other than memory limitation) between the two choices. Thus, considering the proximity constraint may annihilate the power of two choices even in large memory cases as demonstrated in the following example.

Example 4 ($M = K$ and $r = 1$)**.** *In this example, when a request arrives at a server, the server chooses two random choices among itself and its neighbours. Then the request is allocated to the one with lesser load. Since there exists a server at which $\max_i D_i = \Theta(\log n/ \log\log n)$ requests arrive, maximum load of network (i.e., $L = \max_i T_i$) will be at least $\Theta(\log n/ \log\log n)/5$.*

Thus, similar to Question 1 regarding the memory limitation effect, one can pose the following question regarding proximity principle.

Question 2. *In view of the proximity constraint of Scheme II, what are the regimes (in terms of problem parameters) one can benefit from the power of two choices to balance out the load?*

In order to completely analyze load balancing performance of Scheme II, one should consider both sources of correlation simultaneously (which is not the case in above examples). To this end, in the following, we investigate two memory regimes, namely $M = K$ and $M = n^{\alpha}$, for some $0 < \alpha < 1/2$.

Our main result for $M = n^{\alpha}$ is presented in the following theorem.

Theorem 4. *Suppose that $0 < \alpha, \beta < 1/2$ are two constants and let $K = n$, $M = n^{\alpha}$, and $r = n^{\beta}$. Then if*

$$\alpha + 2\beta \geq 1 + 2\log\log n/ \log n,$$

under the Uniform popularity distribution, Strategy II achieves maximum load $L = \Theta(\log\log n)$ and communication cost $C = \Theta(r)$ w.h.p.

Remark 3. *To have a more accessible proof, in Theorem 4, we have assumed that $K = n$. Note that the proof techniques can also be extended to the case where $K = O(n)$.*

In order to prove the theorem, let us first present an interesting result that was shown in [10] as follows.

Theorem 5 ([10]). *Given an almost Δ-regular graph*[4] *G with $e(G)$ edges and n nodes representing n bins, if n balls are thrown into the bins by choosing a random edge with probability at most $O(1/e(G))$ and placing into the smaller of the two bins connected by the edge, then the maximum load is $\Theta(\log\log n) + O\left(\frac{\log n}{\log(\Delta/\log^4 n)}\right) + O(1)$ w.h.p.*

Remark 4. *Note that in the original theorem presented in [10], it is assumed that each edge is chosen uniformly among all edges of graph G. However, here we slightly generalize the result so that each edge is chosen with probability at most $O(1/e(G))$. The proof follows the original proof's idea with some modifications in computation parts, where due to lack of space we omit.*

In order to apply Theorem 5, we first need to define a new graph H as follows.

Definition 4 (Configuration Graph)**.** *For given parameter r, configuration graph H is defined as a graph whose nodes represent the servers and two nodes, say u and v, are connected if and only if u and v have cached a common file and $d(u, v) \leq 2r$ in the torus.*

For every two servers u and v, let $T(u, v)$ be the set of distinct files that have been cached in both nodes u and v. Also denote $|T(u, v)|$ by $t(u, v)$. Define $t(u)$ to be the number of distinct cached files in u. Now, let us define *goodness* of a placement strategy as follows.

Definition 5 (Goodness Property)**.** *For every positive constant $\delta \in [0, 1]$ and $\mu = O(1)$, we say the file placement strategy is (δ, μ)-good, if for every u and v, $t(u) \geq \delta M$ and $t(u, v) < \mu$.*

Lemma 2. *The proportional cache placement strategy introduced in Section II, is (δ, μ)-good w.h.p. for some parameters δ and μ.*

Proof. Clearly, every set of cached files in every node (with replacement) can be one-to-one mapped to a non-negative integral solution of equation $\sum_{i=1}^{K} x_i = M$, where each x_i expresses the number of times that file i has been cached in the node. A combinatorial argument shows that, the equation has $\binom{K+M-1}{M}$ non-negative integer solutions. So for each $1 \leq s \leq M$, we have

$$\mathbf{Pr}\left[t(u) = s\right] = \frac{\binom{K}{s}\binom{M-1}{M-s}}{\binom{K+M-1}{M}}, \tag{2}$$

where we first fix a set of file indexes of size s, say $I = \{i_1, i_2, \ldots, i_s\}$, and then count the number of integral solutions to the equation $\sum_{i\in I} x_i = M - s$.

In order to bound (2), we note that for every $1 \leq a \leq b$, $(b/a)^a \leq \binom{b}{a} \leq b^a$ and also $\binom{b}{a} \leq 2^b$. Recall that we assumed $K = n$ and $M = n^{\alpha}$, $0 < \alpha < 1/2$. Hence for every $1 \leq s \leq \delta M$, we have

$$\begin{aligned}\mathbf{Pr}\left[t(u) = s\right] &\leq \frac{K^s 2^M}{\binom{K}{M}} \leq \frac{K^s 2^M}{(K/M)^M} = (2M)^M K^{s-M}\\ &\leq (2n^{\alpha} n^{\delta-1})^M.\end{aligned}$$

Thus, by choosing $\delta = (1-\alpha)/3$, for every $1 \leq s \leq \delta M$, we have

$$\begin{aligned}\mathbf{Pr}\left[t(u) = s\right] &\leq (2n^{\alpha+\delta-1})^M = (2n^{2\alpha/3-2/3})^M\\ &\leq (2n^{-1/3})^M = n^{-\omega(1)},\end{aligned}$$

where the last equality follows due to $M = n^{\alpha} = \omega(1)$. Now the union bound over all $1 \leq s \leq \delta M$ and n nodes yields

$$\mathbf{Pr}\left[\exists u \in V : t(u) \leq \delta M\right] = n^{-\omega(1)}. \tag{3}$$

By a similar argument, for each $1 \leq t \leq M$ and every u and v, we have

$$\mathbf{Pr}\left[t(u, v) \geq t\right] = \binom{K}{t}\left(\frac{\binom{K+M-t-1}{M-t}}{\binom{K+M-1}{M}}\right)^2.$$

Thus, for any constant $\mu \geq 5/(1-2\alpha)$, we can write

$$\begin{aligned}&\mathbf{Pr}\left[t(u, v) \geq \mu\right]\\ &\leq K^{\mu}\left(\frac{(K+M-\mu-1)!M!}{(K+M-1)!(M-\mu)!}\right)^2\\ &\leq K^{\mu}\left(\frac{M^{\mu}}{K^{\mu}}\right)^2 \leq \frac{M^{2\mu}}{K^{\mu}} = n^{(2\alpha-1)\mu} = O(1/n^5).\end{aligned}$$

By applying the union bound over all pairs of servers, for every u and v we have

$$\mathbf{Pr}\left[t(u, v) \geq \mu\right] = O(1/n^3). \tag{4}$$

Hence, $t(u, v) < \mu$ w.h.p. Putting inequalities (3) and (4) together concludes the proof. □

The following lemma presents some useful properties of H and Strategy II.

Lemma 3. *Conditioning on goodness of file placement and assuming $K = n$, $M = n^{\alpha}$ and $r = n^{\beta}$ with $\alpha + 2\beta \geq 1 + 2\log\log n/\log n$, we have*

(a) *W.h.p. H is almost Δ-regular with $\Delta = \Theta\left(\frac{M^2 r^2}{K}\right)$.*
(b) *For each request, Strategy II samples an edge of H (two servers) with probability $O(1/e(H))$.*

Proof. Consider arbitrary node u with s distinct files. Then by definition of H, for every node v we have

$$\begin{aligned}p_s := \mathbf{Pr}\left[t(u, v) \geq 1 | t(u) = s\right] &= 1 - \left(\frac{K-s}{K}\right)^M\\ &= \frac{sM}{K}(1 + o(1)),\end{aligned}$$

where $1 \leq s \leq M$. On the other hand u and v are connected in H, if in addition $d_G(u, v) \leq 2r$. Therefore for every given node u with s distinct cached files, $d(u)$ in H (degree of

[4] A graph is said to be almost Δ-regular, if each vertex has degree $\Theta(\Delta)$.

u in H) has a binomial distribution $\mathrm{Bin}(b_{2r}(u), p_s)$, where $b_{2r}(u) = |B_{2r}(u)|$. Hence applying a Chernoff bound implies that with probability $1 - n^{-\omega(1)}$, we have

$$d(u) = \frac{sMb_{2r}(u)}{K}(1 + o(1)).$$

Conditioning on the goodness of file placement, $s = t(u) = \Theta(M)$. Also by symmetry of torus, we have $b_{2r}(u) = \Theta(r^2)$, for every u. So, with high probability for every u, we have

$$d(u) = \Theta\left(M^2r^2/K\right),$$

where this concludes the proof of part (a).

Now it remains to show that Strategy II picks an edge of H, with probability $O(1/e(H))$. First, notice that

$$e(H) = \Theta\left(nM^2r^2/K\right) = \Theta(M^2r^2), \tag{5}$$

as $K = n$. Then recall that each file is cached in every node with probability $p = 1 - (1 - 1/K)^M = M/K(1 + o(1))$, independently. For any given node u and file W_j, let $F_j(u)$ be the number of nodes at distance at most r that have cached file W_j. Then $F_j(u)$ has a binomial distribution $\mathrm{Bin}(b_r(u), p)$, where $b_r(u) = |B_r(u)|$. So

$$\mathbf{E}\left[F_j(u)\right] = b_r(u) \cdot p = \Theta(r^2M/K),$$

where $b_r(u) = \Theta(r^2)$ for every u. Since $\alpha + 2\beta \geq 1 + 2\log\log n/\log n$ we have $\mathbf{E}\left[F_j(u)\right] = \omega(\log n)$, for every u and j. Now, applying a Chernoff bound for $F_j(u)$ implies that with probability $1 - n^{-\omega(1)}$, $F_j(u)$ concentrates around its mean and hence, w.h.p., we have for every u and j

$$F_j(u) = \Theta(r^2M/K) = \Theta(r^2M/n).$$

Consider an edge $(u, v) \in E(H)$, with $t(u, v) = t$. Define $S_{u,v}$ to be the set of nodes that may pick pair u and v randomly in Strategy II. It is not hard to see that $|S_{u,v}| = O(r^2)$. Now we have,

$$\begin{aligned}
&\mathbf{Pr}\left[(u, v) \in E(H) \text{ is picked by Strategy II} | t(u, v) = t\right] \\
&\quad = \sum_{j \in T(u,v)} \frac{1}{K} \sum_{w \in S_{u,v}} \frac{1}{n} \frac{1}{\binom{F_j(w)}{2}} \\
&\quad = \frac{1}{n^2} \sum_{j \in T(u,v)} \sum_{w \in S_{u,v}} \frac{1}{\binom{F_j(w)}{2}} \\
&\quad = \frac{1}{n^2} \sum_{j \in T(u,v)} \sum_{w \in S_{u,v}} \Theta(n^2/r^4M^2). \qquad (6)
\end{aligned}$$

Conditioned on "goodness," we have for every $(u, v) \in E(H)$, $1 \leq t(u, v) < \mu$. So (6) can be simplified as

$$\begin{aligned}
&\mathbf{Pr}\left[(u, v) \in E(H) \text{ is picked by Strategy II}\right] \\
&\quad \leq \Theta(\mu|S_{u,v}|/r^4M^2) \\
&\quad = O(1/r^2M^2) = O(1/e(H)),
\end{aligned}$$

where the last equality follows from (5). □

Proof of Theorem 4. Applying Lemma 3 shows that w.h.p. the configuration graph H is an almost Δ-regular graph where $\Delta = M^2r^2/n$. Moreover, in each step, every edge of H is chosen randomly with probability $O(1/e(H))$. Hence, we can apply Theorem 5 and conclude that w.h.p. the maximum load is at most

$$\Theta(\log\log n) + O\left(\frac{\log n}{\log(\Delta/\log^4 n)}\right) = \Theta(\log\log n) + O(1),$$

where it follows because $\alpha + 2\beta \geq 1 + 2\log\log n/\log n$ and hence $\Delta = M^2r^2/n = n^{2\alpha+2\beta-1} > n^{\alpha}$. □

Now let us present our next result regarding to the $M = K$ regime.

Theorem 6. *Suppose $M = K$ and Uniform distribution $\mathcal{P}$ over the file library. Then Strategy II achieves the maximum load $L = \Theta(\log\log n)$ and communication cost $C = \Theta\left(n^{\beta}\right)$ for any $\beta = \Omega(\log\log n/\log n)$.*

Proof. Let us choose $r = n^{\beta}$, for some $\beta = \Omega(\log\log n/\log n)$. By the assumption $M = K$, the configuration graph H (corresponding to r) is a graph in which two nodes u and v are connected if and only if $d(u, v) \leq 2r$. Since our network is symmetric, for every u, $|B_r(u)| = \Theta(r^2)$ and hence H is a regular graph with $\Delta = \Theta(r^2)$. Also it is not hard to see that Strategy II is equivalent to choosing an edge uniformly from H. Applying Theorem 5 ([10]) to H results in the maximum load of $\Theta(\log\log(n))$. In addition, choosing two random nodes in $|B_r(u)| = \Theta(r^2)$ results in communication cost of $C = \Theta(r) = \Theta\left(n^{\beta}\right)$. □

The main point of Theorem 6 is that we can just have $C = \Theta\left(n^{\beta}\right)$, for $\beta = \Omega(\log\log n/\log n)$, to benefit from the luxury of power of two choices, which is a very encouraging result.

V. Simulations

In this section, we demonstrate the simulation results for two strategies introduced in the previous sections, namely *nearest replica* and *proximity aware two choices*. The simulation results are shown for the torus topology. Here, we consider Uniform popularity over the file library. As a result, the file placement is also considered to be uniform over the servers' storage.

Figure 1 shows the maximum load of Strategy I as a function of the number of servers where different curves correspond to different cache sizes. The network graph is a torus, where 100 files with Uniform popularity are placed uniformly at random in each node. Each point is an average of 10000 simulation runs. This figure confirms that the logarithmic growth of the maximum load, asymptotically proved in Theorem 1, also holds for intermediate values of $n \approx 100, \ldots, 3000$ which makes the result of Theorem 1 more general. Comparing different curves reveals the fact that in larger cache size setting, we have a more balanced network. That happens because enlarging cache sizes results in a more uniform Voronoi tessellation, i.e., having cells with smaller variation in size.

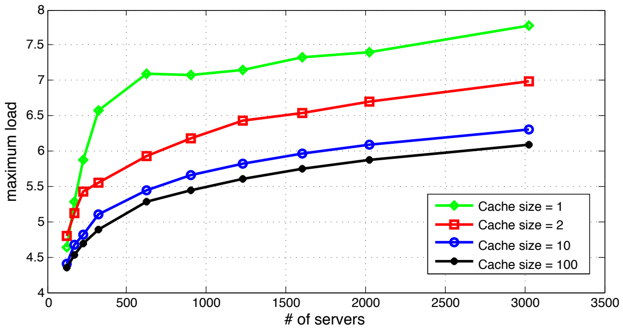

Fig. 1: The maximum load versus number of servers for Strategy I.

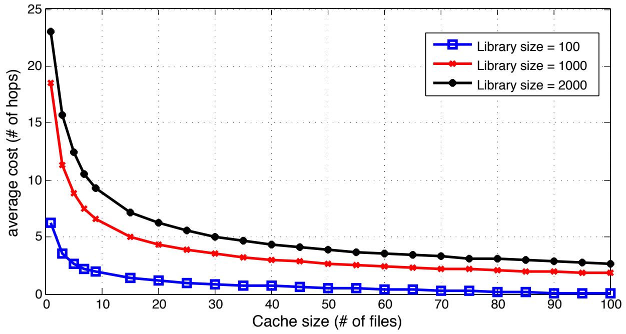

Fig. 2: The communication cost versus cache size for Strategy I.

Furthermore, Figure 2 shows the communication cost of Strategy I as a function of cache size where different curves correspond to different library sizes. Here, the network graph is a torus of size 2025 and each point is an average of 10000 simulation runs. This figure is in agreement with the result of Theorem 3.

In order to simulate Strategy II, first we set $r = \infty$ to study the effect of cache size on the maximum load and communication cost and then consider the effect of limited r on the performance of the system. Figure 3 shows the maximum load of the network versus number of servers where each curve demonstrates a different cache size. The network graph is a torus, where 2000 files with Uniform popularity are placed uniformly at random in each node. Each point is an average of 800 simulation runs. In each curve, since cache size and number of files are fixed, increasing the number of servers translates to increasing each file replication.

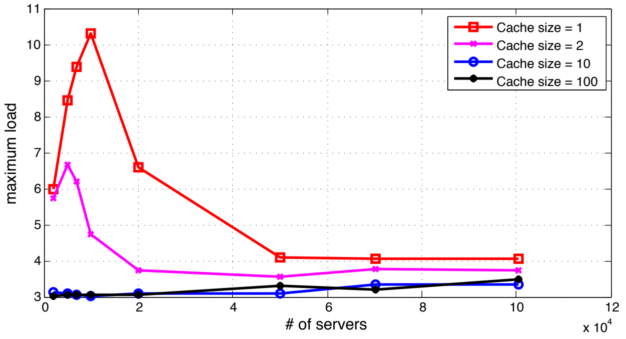

Fig. 3: The maximum load versus number of servers for Strategy II. Here, we assume $r = \infty$.

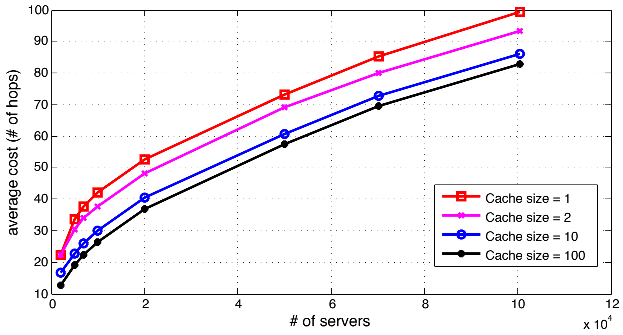

Fig. 4: The communication cost versus number of servers for Strategy II. Here, we assume $r = \infty$.

In Figure 3, when the file replication is low, due to high correlation between the two choices of Strategy II, power of two choices is not expected. This is reflected in Figure 3; for example in the curve corresponding to $M = 1$ for $n \leq 10000$ we have a fast growth in maximum load which mimics the load balancing performance of Strategy I. However, for $n > 50000$, since there is enough file replication in the network, the load balancing performance is greatly improved due to power of two choices. This is in accordance with the lessons learned from Section IV. Also, for $10000 < n < 50000$, we have a transition region where a mixed behaviour is observed. Likewise, the curve for $M = 2$ shows a similar trend. However, for $M = 10$ due to memory abundance, we only observe the latter behaviour where power of two choices is achieved. Observations made above from Figure 3 has an important practical implication. Since employing Strategy II is only beneficial in networks with high file replication, for other situations with limited cache size, the less sophisticated Strategy I is a more proper choice.

Figure 4 draws the communication cost versus number of servers for various cache sizes for similar setting used in Figure 3. Since in this figure there is no constraint on the proximity the communication cost growth is of order $\Theta(\sqrt{n})$.

In simulations depicted in Figures 3 and 4, we only consider the case $r = \infty$. In order to investigate the effect of parameter r on the performance of the system, in Figure 5, we have simulated network operation for different values of r. This results in a trade-off between the maximum load and communication cost, as shown in Figure 5. Here we consider a torus with 2025 servers, where 500 files with Uniform popularity are placed uniformly at random in each node. Each point is an average

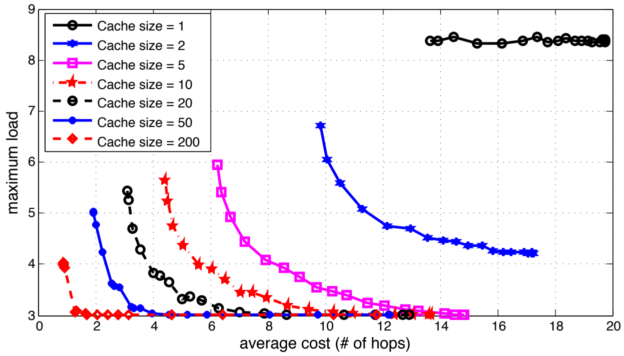

Fig. 5: The tradeoff between the maximum load and communication cost for Strategy II.

of 5000 simulation runs.

In this figure, like before, i.e., Figure 3, we observe two performance regimes based on the cache size M. In high memory regime, e.g., for curves corresponding to $M = 50$ and $M = 200$, we can achieve the power of two choices by sacrificing a negligible communication cost. On the other hand, in low memory regime, i.e., $M = 1$, we cannot decrease the maximum load even at the expense of high communication cost values. For intermediate values of M, we clearly observe the trade-off between the maximum load and communication cost.

VI. Discussion, Open Questions and Future Directions

In this section, first, we summarize the paper. Then we bring forward discussion about the proposed schemes, open questions and possible future directions.

In summary, we have considered the problem of randomized load balancing and its tension with communication cost in cache networks. By proposing two request assignment schemes, the trade-off between communication cost and maximum load has been investigated analytically. Moreover, simulation results support our theoretical findings and provide practical design guidelines.

The proposed *proximity-aware two choices* scheme can be implemented in a distributed manner. To see why, notice that upon arrival of each request at each server, this strategy needs two kinds of information to redirect the request. This information can be provided to the requesting server without the need for a centralized authority in the following way. The first one is the cache content of other users in its neighborhood with radius r. Since, the cache content dynamic of servers is much slower than the requests arrival, this can be done by periodic polling of nearby servers without introducing much overhead. Also, the cache content placement at each server can be implemented via efficient Distributed Hash Table (DHT) schemes (see, e.g., [30] and [31]), which can be adopted to dynamic library popularity profiles. This will also enable all users to obtain global cache content information in a robust and distributed manner. In this paper we assume a static profile and do not go into the details of such schemes. The second type of information is the queue length information of two randomly chosen nodes inside its neighborhood with radius r, which can also be efficiently done in a distributed manner by polling or piggybacking.

In practice, request arrivals and servers' operation happen in continuous time which needs a queuing theory based performance analysis. However, as shown in [6] and [32], the behaviour of load balancing schemes in continuous time (i.e., known as the supermarket model) and static balls and bins problems are closely related. Thus, we conjecture that our proposed scheme will also have the same performance in queuing theory based model. We postpone a rigorous analysis of such scenario to future work.

In this paper we do not consider any form of coding in the cache content placement and content delivery phases. However, as recently shown in [33] (and follow up works [34], [35], [36]), employing coding in cache networks can reduce network traffic dramatically. An important future work will be investigating the effect of coding techniques in the context of our proposed randomized load balancing scheme.

Acknowledgment

The authors would like to thank Seyed Abolfazl Motahari, Omid Etesami, Thomas Sauerwald and Farzad Parvaresh for helpful discussions and feedback.

References

[1] Cisco, “Cisco visual networking index: global mobile data traffic forecast update, 20132018,” *White Paper*, 2014.

[2] G. Zhang, Y. Li, and T. Lin, “Caching in information centric networking: A survey,” *Computer Network*, vol. 57, pp. 3128–3141, 2013.

[3] E. Nygren, R. K. Sitaraman, and J. Sun, “The akamai network: a platform for high-performance internet applications,” *Operating Systems Review*, vol. 44, no. 3, pp. 2–19, 2010.

[4] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, “Femtocaching: Wireless video content delivery through distributed caching helpers,” in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 1107–1115.

[5] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal, “Balanced allocations,” *SIAM J. Comput.*, vol. 29, no. 1, pp. 180–200, 1999.

[6] M. Mitzenmacher, “The power of two choices in randomized load balancing,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, no. 10, pp. 1094–1104, 2001.

[7] M. Adler, S. Chakrabarti, M. Mitzenmacher, and L. E. Rasmussen, “Parallel randomized load balancing,” *Random Struct. Algorithms*, vol. 13, no. 2, pp. 159–188, 1998.

[8] C. Lenzen and R. Wattenhofer, “Tight bounds for parallel randomized load balancing,” *Distributed Computing*, vol. 29, no. 2, pp. 127–142, 2016.

[9] P. Berenbrink, A. Czumaj, A. Steger, and B. Vöcking, “Balanced allocations: The heavily loaded case,” *SIAM J. Comput.*, vol. 35, no. 6, pp. 1350–1385, 2006.

[10] K. Kenthapadi and R. Panigrahy, “Balanced allocation on graphs,” in *Proc. 17th Symp. Discrete Algorithms (SODA)*, 2006, pp. 434–443.

[11] G. Peng, “CDN: content distribution network,” *CoRR*, vol. cs.NI/0411069, 2004. [Online]. Available: http://arxiv.org/abs/cs.NI/0411069

[12] M. Roussopoulos and M. Baker, “Practical load balancing for content requests in peer-to-peer networks,” *Distributed Computing*, vol. 18, no. 6, pp. 421–434, 2006. [Online]. Available: http://dx.doi.org/10.1007/s00446-005-0150-7

[13] M. Leconte, M. Lelarge, and L. Massoulié, "Bipartite graph structures for efficient balancing of heterogeneous loads," in *ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '12, London, United Kingdom, June 11-15, 2012*, 2012, pp. 41–52.

[14] S. Manfredi, F. Oliviero, and S. P. Romano, "A distributed control law for load balancing in content delivery networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 1, pp. 55–68, 2013.

[15] F. Xia, A. M. Ahmed, L. T. Yang, and Z. Luo, "Community-based event dissemination with optimal load balancing," *IEEE Trans. Computers*, vol. 64, no. 7, pp. 1857–1869, 2015.

[16] C. Chen, Y. Ling, M. Pang, W. Chen, S. Cai, Y. Suwa, and O. Altintas, "Scalable request routing with next-neighbor load sharing in multi-server environments," in *19th International Conference on Advanced Information Networking and Applications (AINA 2005), 28-30 March 2005, Taipei, Taiwan*, 2005, pp. 441–446.

[17] Y. Xia, A. Dobra, and S. C. Han, "Multiple-choice random network for server load balancing," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 6-12 May 2007, Anchorage, Alaska, USA*, 2007, pp. 1982–1990.

[18] V. Shah and G. de Veciana, "High-performance centralized content delivery infrastructure: Models and asymptotics," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1674–1687, 2015.

[19] C. Liu, R. K. Sitaraman, and D. Towsley, "Go-with-the-winner: Performance based client-side server selection," in *2016 IFIP Networking Conference, Networking 2016 and Workshops, Vienna, Austria, May 17-19, 2016*, 2016, pp. 404–412.

[20] A.-M. K. Pathan, C. Vecchiola, and R. Buyya, "Load and proximity aware request-redirection for dynamic load distribution in peering cdns," in *On the Move to Meaningful Internet Systems: OTM 2008, OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, Monterrey, Mexico, November 9-14, 2008, Proceedings, Part I*, 2008, pp. 62–81.

[21] J. Tang, W.-P. Tay, and Y. Wen, "Dynamic request redirection and elastic service scaling in cloud-centric media networks," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1434–1445, 2014.

[22] R. Stanojevic and R. Shorten, "Load balancing vs. distributed rate limiting: An unifying framework for cloud control," in *Proceedings of IEEE International Conference on Communications, ICC 2009, Dresden, Germany, 14-18 June 2009*, 2009, pp. 1–6.

[23] P. Berenbrink, A. Brinkmann, T. Friedetzky, and L. Nagel, "Balls into bins with related random choices," *J. Parallel Distrib. Comput.*, vol. 72, no. 2, pp. 246–253, 2012.

[24] B. Godfrey, "Balls and bins with structure: balanced allocations on hypergraphs," in *Proc. 19th Symp. Discrete Algorithms (SODA)*, 2008, pp. 511–517.

[25] A. Pourmiri, "Balanced allocation on graphs: A random walk approach," in *Computing and Combinatorics - 22nd International Conference, COCOON 2016, Ho Chi Minh City, Vietnam, August 2-4, 2016, Proceedings*, 2016, pp. 330–341.

[26] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: evidence and implications," in *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, Mar 1999, pp. 126–134 vol.1.

[27] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 1–14.

[28] A. Pourmiri, M. Jafari Siavoshani, and S. P. Shariatpanahi, "Proximity-aware balanced allocations in cache networks," *CoRR*, vol. abs/1610.05961, 2016. [Online]. Available: http://arxiv.org/abs/1610.05961

[29] L. Devroye, "The expected length of the longest probe sequence for bucket searching when the distribution is not uniform," *J. Algorithms*, vol. 6, no. 1, pp. 1–9, 1985.

[30] D. Bauer, P. Hurley, and M. Waldvogel, "Replica placement and location using distributed hash tables," in *32nd Annual IEEE Conference on Local Computer Networks (LCN 2007), 15-18 October 2007, Clontarf Castle, Dublin, Ireland, Proceedings*, 2007, pp. 315–324.

[31] D. R. Karger, E. Lehman, F. T. Leighton, R. Panigrahy, M. S. Levine, and D. Lewin, "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web," in *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, 1997, pp. 654–663.

[32] M. Mitzenmacher, A. W. Richa, and R. Sitaraman, "The power of two random choices: A survey of technique and results," *In Handbook of Randomized Computation Volume 1*, pp. 255–312, 2001.

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

[34] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Information Theory*, vol. 62, no. 2, pp. 849–869, 2016.

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

[36] S. P. Shariatpanahi, A. S. Motahari, and B. H. Khalaj, "Multi-server coded caching," *CoRR*, vol. abs/1503.00265, 2015.