

Storage, Communication, and Load Balancing Trade-off in Distributed Cache Networks

Mahdi Jafari Siavoshani, *Member, IEEE*, Ali Pourmiri, Seyed Pooya Shariatpanahi

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. In this framework, we investigate the tension between storage resources, communication cost, and load balancing performance. To this end, we propose a randomized load balancing scheme which simultaneously considers cache size limitation and proximity in the server redirection process.

In contrast to the classical power of two choices setup, 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. However, we prove that in certain regimes of problem parameters, our scheme results in the maximum load of order $\Theta(\log \log n)$ (here n is the network size). This is an exponential improvement compared to the scheme which assigns each request to the nearest available replica. Interestingly, the extra communication cost incurred by our proposed scheme, compared to the nearest replica strategy, is small. Furthermore, our extensive simulations show that the trade-off trend does not depend on the network topology and library popularity profile details.

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 [2]. 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. [3], [4], and mobile video delivery [5]. 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.

The authors' names appear in alphabetical order.

This work has been presented in part at IPDPS 2017, [1].

M. Jafari Siavoshani is with the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran (email: mjafari@sharif.edu).

A. Pourmiri is with the Department of Computer Engineering, University of Isfahan, Isfahan, Iran (email: a.pourmiri@comp.ui.ac.ir).

S. P. Shariatpanahi is with the School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran (email: pooya@ipm.ir).

This work has been supported in part by IPM.

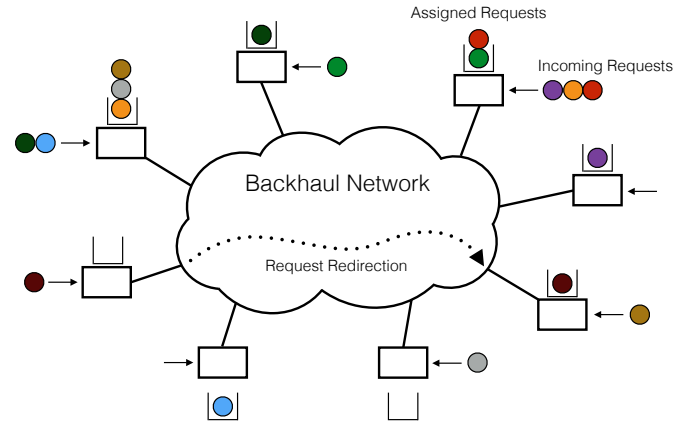


Fig. 1: A general distributed cache network.

As a schematic view of a typical cache network see Figure 1. This figure shows caching servers connected through a backhaul network. These servers are responsible for delivering contents requested by customers. Each server should assign the demand to a responding server (which could be itself) via an assignment strategy. In every cache network we have three critical parameters, namely, *Storage Resource*, *Communication Cost*, and *Network Imbalance*. Storage, or memory characterizes what percentage of the total content library can be cached at each server. Communication cost is the amount of data transferred inside the backhaul network to satisfy content requests. Finally, network imbalance characterizes how uniformly the requested contents' loads are distributed among different responding servers. This is usually measured by comparing the load of the busiest server with the average load of all servers after request assignments. Every request assignment strategy, in fact, leads to a trade-off between these three quantities.

From a practical viewpoint, we are interested in *distributed* server selection strategies which are scalable in large networks. Randomized load balancing via the so-called “power of two choices” is a well-investigated paradigm in parallel and distributed settings [6], [7], [8], [9]. In this approach, upon arrival of a request, the corresponding user will query about the current loads of two independently at random chosen servers, and then allocates the request to the least loaded server. Considering only the load balancing perspective, Berenbrink et al., [10], showed that in this scheme after allocating m balls (requests, tasks, etc.) to n bins (servers, machines, etc.) 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.

Although the power of two choices strategy addresses the load balancing issue in a distributed manner, it does not consider the role of two other important quantities, namely, memory and communication cost. Our goal in this paper is to extend the power of two choices framework in order to characterize the trade-off between these three quantities. Our results show that the maximum load, communication cost, and servers' memory are three entangled parameters. Thus, the previous studies are not sufficient for designing a practical load balancing strategy in cache networks.

B. Problem Setting and Our Contributions

In this paper, we consider a general cache network model that entails basic characteristics of many practical scenarios. We consider a network of n servers and a library of size K files. Each server can cache M files in network low-traffic hours. Let us assume a popularity distribution $\mathcal{P} = \{p_1, \dots, p_K\}$ on the library. We assume that the cache content placement at each server is proportional to this popularity distribution. In high-traffic hours there are n sequential file requests, from the library, distributed among servers uniformly at random. 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* L of an assignment scheme as the maximum number of allocations to any single server after assigning all requests. The *communication cost* C is the average number of hops required to deliver the requested file to its request origin.

As the baseline assignment scheme, we consider that each request arrived at every server is dispatched to the nearest file replica. This scheme results in the minimum communication cost, while ignoring maximum load of servers. We show that, for a grid topology and 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 L in the interval $[\Omega(\log n / \log \log n), O(\log n)]$ with high probability¹ (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 incurred in this scheme for Uniform and Zipf popularity distributions. In particular, we derive the communication cost C of order $\Theta(\sqrt{K/M})$ for the Uniform distribution in a grid topology.

In contrast, we propose a new scheme which considers memory, maximum load, and communication cost, 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 constraint, 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, for grid topology, 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, if we set $\beta = \frac{1-\alpha}{2} + \log \log n / \log n$ we achieve the power of two choices with the communication cost of order $\Theta(r) = \Theta\left(n^{\frac{1-\alpha}{2}} \log n\right)$. This communication cost is only $\log n$ factor above the communication cost achieved by the nearest replica strategy, which is $\Theta(\sqrt{K/M}) = \Theta(n^{\frac{1-\alpha}{2}})$. This result also shows an interesting trade-off between memory and communication cost. Simply put, if memory is decreased by decreasing α , then we should increase β such that $\alpha + 2\beta$ remains more than 1, and then we can achieve the power of two choices. In a similar manner, if we cannot tolerate high communication costs, we should increase memory size such that $\alpha + 2\beta$ remains more than 1 (assuming that both are less than $1/2$). Furthermore, it should be noted that while our theoretical results are derived for grid networks, the main reason for assuming a grid is presentation clarity and the results can be extended to other topologies.

While our theoretical results are derived for large networks (i.e., asymptotic analysis), our simulation results show their validity even for finite sized networks. Also, in simulations we investigate the problem in more diverse settings such as considering other network topologies.

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], [8], and [15]). Randomized load balancing via the power of two choices, is a popular approach in this direction [7]. 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 to these works, we follow a theoretical approach to derive provable results for cache networks.

Theoretical works investigating the power of two choices in cache networks all just consider the role of two parameters among *Memory Resource*, *Load Balancing*, and *Communication Cost*. These works can be summarized in three categories as follow.

Memory Resource vs. *Communication Cost* trade-off has been investigated in many works such as [5], [18], [19], [20], and [21]. None of these papers have considered the load balancing issue in cache networks, which is very important in practice.

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

Load Balancing vs. Memory Resource trade-off has been investigated in [22] and [23]. In [22] the authors consider the supermarket model for performance evaluation of CDNs. Although the work [22] considers the memory limitation into account, it does not consider the proximity principle which is a central issue in our paper. Liu et al. [23] 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.

Communication Cost vs. Load Balancing trade-off has been investigated in [24], [25], [26], [27], [28], [29], and [30], without considering the effect of cache size limitation. Although the works [24], [25], and [26] have mentioned this trade-off, non of them provides a rigorous analysis.

In contrast to the standard balls and bins model, the works [27], [28], [29], and [30] introduced the effect of proximity constraint to the ball and bins framework. 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 [6]. However, in many settings, selecting any two random servers might be infeasible or costly. In other words, this proximity constraint translates to a *correlation* between the two choices, i.e., the balls and bins model with *related choices*.

The most related work to our paper is [28]. Kenthapadi and Panigrahi [28] proposed a model where n bins are connected 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 $\Theta(\log \log n)$ w.h.p.

Although the model used in [28] considers the proximity principle by assigning each request to the origin neighbors, it cannot be directly applied to our cache network setup. First, they do not consider multi-hop communication, while in practice the communication is done in a multi-hop fashion. Second, the above model cannot accommodate the cache size limitation of servers. Cache size limitation introduces the notion of cache content placement which should be based on the popularity profile. In addition, this limitation will introduce a new source of correlation between choices which is not considered in [28].

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*, is investigated as the baseline scheme. In Section IV, we propose and analyze the *proximity-aware two choices strategy*, which at the same time considers memory limitation, proximity constraint, and benefits from the power of two choices. In Section V the performance of these two schemes are investigated via extensive simulations. Finally, our discussions and concluding remarks are presented in Section VI.

II. NOTATION AND PROBLEM SETTING

A. Notation

Throughout the paper, 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 for 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 $\text{Po}(\lambda)$ 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. 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 nodes. 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 $\mathcal{W} = \{W_1, \dots, W_K\}$, whereas the popularity profile follows a known distribution $\mathcal{P} = \{p_1, \dots, p_K\}$.

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 $\mathcal{P} = \{p_1, \dots, p_K\}$ with replacement, independent of other nodes.

Remark 2. *It should be noted that random caching with replacement is less efficient compared to random caching without replacement. However, almost all our results in the paper are for the memory sizes which satisfy $M \ll K$. In this regime, caching the same file more than once in each server is very improbable and the two cache content placement strategies will be similar in large networks. In all other places where $M \ll K$ does not hold, we use random cache content placement without replacement.*

Consider a time block during which n files are requested from the servers sequentially that are chosen uniformly at random. Let D_i denotes the number of requests (demands) arrived at server i . Then for large n we have $D_i \sim \text{Po}(1)$ 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

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

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

$$p_i = \frac{1/i^\gamma}{\sum_{j=1}^K 1/j^\gamma}, \quad i = 1, \dots, K,$$

which has been confirmed to be the case in many practical applications [31], [32].

For any given cache content placement, an assignment strategy determines how each request is mapped to a server. Let T_i denotes 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 $r = O(\sqrt{K \log n/M})$. 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 of Lemma 1. Upper Bound – Fix a node u and w.l.o.g. assume that u is denoted by pair $(0, 0)$ in the torus. With

respect to u and some positive number $r > 0$ define four areas as follows

$$\begin{aligned} A_1(u) &:= \{(x, y) : 0 \leq y \leq x/2 \text{ and } (x, y) \in B_r(u)\}, \\ A_2(u) &:= \{(x, y) : 0 \leq -x \leq y/2 \text{ and } (x, y) \in B_r(u)\}, \\ A_3(u) &:= \{(x, y) : 0 \leq -y \leq -x/2 \text{ and } (x, y) \in B_r(u)\}, \\ A_4(u) &:= \{(x, y) : 0 \leq x \leq -y/2 \text{ and } (x, y) \in B_r(u)\}, \end{aligned}$$

which are shown in Fig. 2. It is easy to see that all four areas have the same size, that is

$$\begin{aligned} |A_1(u)| &= \sum_{y=0}^{\lfloor r/3 \rfloor} \sum_{x=2y}^{r-y} 1 \\ &= \sum_{y=0}^{\lfloor r/3 \rfloor} (r - 3y + 1) \\ &\geq \sum_{y=0}^{\lfloor r/3 \rfloor} 3y \\ &\geq r^2/8. \end{aligned} \tag{1}$$

Let us fix some arbitrary $1 \leq j \leq K$ and for every node u define indicator random variable $X_{u,j}$ taking value 1 if u has cached file W_j and there is no node in $A_1(u)$ that has cached file W_j , and 0 otherwise. Then,

$$\Pr[X_{u,j} = 1] = \left(1 - \left(1 - \frac{1}{K}\right)^M\right) \left(1 - \frac{1}{K}\right)^{M(|A_1(u)|-1)},$$

where the first term determines the probability that u caches W_j and the second one determines the probability that nodes in $A_1(u) \setminus \{u\}$ do not cache W_j . By setting $r = 5\sqrt{K \log n/M}$ and applying Inequality (1) we have,

$$\begin{aligned} \left(1 - \frac{1}{K}\right)^{M(|A_1(u)|-1)} &= e^{-\frac{25 \log n}{8}} (1 + o(1)) \\ &= O(n^{-3}), \end{aligned} \tag{2}$$

where it follows from $1 - 1/K = e^{-1/K}(1 + o(1))$ and $M/K = o(1)$. Moreover, by using the approximation $1 - (1 - 1/K)^M = M(1 + o(1))/K$, we have

$$\Pr[X_{u,j} = 1] \leq \frac{M(1 + o(1))}{K \cdot n^3}.$$

Therefore applying the union bound over all n nodes and K files implies that w.h.p. for every u there exists at least one node in $A_1(u)$ which shares a common file with u , supported that we choose $r = 5\sqrt{K \log n/M}$. We can similarly prove the same argument for $A_2(u)$, $A_3(u)$ and $A_4(u)$.

Suppose that u has cached file W_j , and we want to find an upper bound for the size of the Voronoi cell centered at u corresponding to W_j . In order to do this let us define $v^1 = (v_x^1, v_y^1) \in A_1(u)$ to be the nearest node to u with file W_j . Similarly define $v^i \in A_i(u)$, $2 \leq i \leq 4$. W.l.o.g. assume that u is the origin and the nodes are in a xy -coordinate system. Define

$$B := \{(x, y) : v_x^3 \leq x \leq v_x^1 \text{ and } v_y^4 \leq y \leq v_y^2\}.$$

Now we show that the Voronoi cell of u is contained in B , and thus the size of B is an upper bound to the size of the

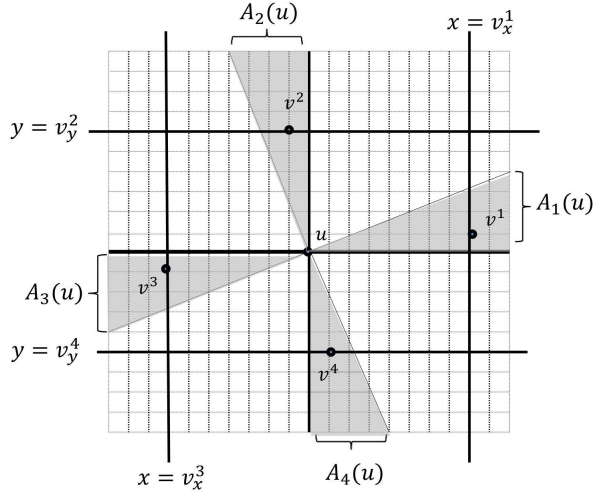


Fig. 2: Demonstration of regions $A_1(u), \dots, A_4(u)$ used in the upper bound proof of Lemma 1.

Voronoi cell. Consider Fig. 2. Let us consider node w in the complement of B with $w_y > v_y^2$ and $w_x > 0$. Assume that P_{uw} is a shortest path from w to u that passes node $(0, v_y^2)$. By definition of $A_2(u)$, we know the length of a shortest path from $(0, v_y^2)$ to (v_x^2, v_y^2) is $|v_x^2| \leq v_y^2/2$. This shows that w is closer to v^2 than u , and by definition it does not belong to the Voronoi cell centered at u . We similarly can show that each arbitrary node $w \in B^c$ is closer to either v^i 's rather than u . Since we arbitrarily choose u and $1 \leq j \leq K$, there is sub-grid B that contains every Voronoi cell in \mathcal{V}_j , centered at any given u . So the size of any Voronoi cell centered at an arbitrary node u is bounded from above by $|B|$, which is at most $4r^2 = O(K \log n/M)$.

Lower Bound – Let us define indicator random variable $Y_{u,j}$ for every u and some fixed j taking value 1 if u has cached file W_j and there is no $v \in B_r(u)$ that has cached file W_j , and 0 otherwise. Note that $|B_r(u) \setminus \{u\}| = 2r(r+1)$. Then we have

$$\Pr[Y_{u,j} = 1] = \left(1 - \left(1 - \frac{1}{K}\right)^M\right) \left(1 - \frac{1}{K}\right)^{M[2r(r+1)]}.$$

By setting $r = \sqrt{\epsilon \cdot K \cdot \log n / 4M}$ and using similar approximations used in (2) we have

$$p \triangleq \Pr[Y_{u,j} = 1] = \frac{M(1 + o(1))}{K \cdot n^{\epsilon/2}}.$$

Let $Y_j = \sum_u Y_{u,j}$. Then, we have the following claim.

Claim 1. For every j we have $Y_j \geq 1$ with probability $1 - o(1)$.

This claim shows that there exists at least a Voronoi cell of size $\Theta(r^2) = \Theta(K \log n/M)$ which concludes the proof.

Now, in order to prove the claim note that

$$\mathbb{E}[Y_j] = \sum_u \mathbb{E}[Y_{u,j}] = n \cdot \frac{M(1 + o(1))}{K \cdot n^{\epsilon/2}} = Mn^{\epsilon/2}(1 + o(1)).$$

Also, we know that

$$\begin{aligned} \text{Var}[Y_j] &= \sum_{u,v} \text{Cov}(Y_{u,j}, Y_{v,j}) \\ &= \sum_{u,v} \mathbb{E}[Y_{u,j}Y_{v,j}] - \mathbb{E}[Y_{u,j}] \mathbb{E}[Y_{v,j}] \\ &= \sum_{u,v} \left(\Pr[Y_{u,j} = 1, Y_{v,j} = 1] \right. \\ &\quad \left. - \Pr[Y_{u,j} = 1] \cdot \Pr[Y_{v,j} = 1] \right), \end{aligned} \quad (3)$$

where the last equality holds because $Y_{u,j}$'s are indicator random variables. It is easy to see that for every u and v with $d_G(u, v) > 2r$, $\text{Cov}(Y_{u,j}, Y_{v,j}) = 0$ as cache content placement at different nodes are independent processes. So we only consider pairs u and v , with $d_G(u, v) \leq 2r$. Then for each pair of nodes three following cases should be considered:

- $u = v$: In this case we have

$$\begin{aligned} \Pr[Y_{u,j} = 1, Y_{v,j} = 1] - \Pr[Y_{u,j} = 1] \cdot \Pr[Y_{v,j} = 1] \\ = \Pr[Y_{u,j} = 1] - p^2 = p(1 - p). \end{aligned} \quad (4)$$

- $0 < d_G(u, v) \leq r$: By definition of indicator random variables $Y_{u,j}$'s, we have

$$\begin{aligned} \Pr[Y_{u,j} = 1, Y_{v,j} = 1] - \Pr[Y_{u,j} = 1] \cdot \Pr[Y_{v,j} = 1] \\ = 0 - p^2. \end{aligned} \quad (5)$$

- $r < d_G(u, v) \leq 2r$: In this case we have

$$\begin{aligned} \Pr[Y_{u,j} = 1, Y_{v,j} = 1] - \Pr[Y_{u,j} = 1] \cdot \Pr[Y_{v,j} = 1] \\ = \Pr[Y_{u,j} = 1 | Y_{v,j} = 1] \Pr[Y_{v,j} = 1] - p^2 \\ \leq \frac{M(1 + o(1))}{K} p - p^2 \leq \frac{2M}{K} p. \end{aligned} \quad (6)$$

Now let us split the summation (3) based on $d_G(u, v)$ as follows

$$\begin{aligned} \text{Var}[Y_j] &= \sum_u \text{Cov}(Y_{u,j}, Y_{u,j}) \\ &\quad + \sum_u \sum_{v: 0 < d_G(u, v) \leq r} \text{Cov}(Y_{u,j}, Y_{v,j}) \\ &\quad + \sum_u \sum_{v: r < d_G(u, v) \leq 2r} \text{Cov}(Y_{u,j}, Y_{v,j}). \end{aligned}$$

Applying results (4-6) yields

$$\begin{aligned} \text{Var}[Y_j] &\leq np(1 - p) - n|B_r(u)|p^2 + n|B_{2r}(u)| \frac{2Mp}{K} \\ &\leq np + 4nr(2r + 1) \frac{2Mp}{K} \\ &\leq np + 6\epsilon np \log n \\ &\leq 7np \log n, \end{aligned}$$

where we use the fact that $4r(2r + 1) \leq 9r^2 \leq 3\epsilon K \log n$. Applying Chebychev's inequality leads to

$$\begin{aligned} \Pr[|Y_j - \mathbb{E}[Y_j]| \geq \mathbb{E}[Y_j]/2] &\leq \frac{4 \text{Var}[Y_j]}{\mathbb{E}[Y_j]^2} \\ &\leq \frac{28np \log n}{n^2 p^2} \\ &= \frac{28 \log n}{np} = o(1). \end{aligned}$$

Therefore, Y_j concentrates around its mean, i.e., $np = \Theta(n^{\epsilon/2})$, which proves the claim. \square

Now, we are ready to present our main results for this section which characterize the maximum load of Strategy I, for two different parameter regimes, 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 the Appendix) 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 exists 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 . \square

Remark 3. 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. For further numerical investigation on this remark refer to Section V.

Theorem 2. Suppose that $K = n$ and $M = n^\alpha$, for some $0 < \alpha < 1/2$. Then, under the Uniform distribution, the maximum load in the asymptotic regime is lower bounded by $\Omega(\log n / \log \log n)$ and is upper bounded by $O(\log n)$ w.h.p.

Proof of Theorem 2. To establish upper bound $O(\log n)$ for the maximum load, we follow the first part of proof of Theorem 1. To obtain a lower bound, consider an arbitrary server u that has cached file set S with s distinct files. Note that by Lemma 2, we have for every node u , $s = \Theta(M)$ with high probability. Let us define the indicator random variable $X_{u,j}$, $W_j \in S$, taking 1 if the nearest replica of W_j is outside of $B_r(u)$, where $r = \sqrt{K/2M}$ and zero otherwise. It is easy to see that $X_{u,j}$'s are correlated. For example, consider the set of files $T = \{W_{j_1}, W_{j_2}, \dots, W_{j_t}\} \subset S$, where $X_{u,j'} = 1$ for every $W_{j'} \in T$. Then conditioned on this event, each node in $B_r(u)$ has cached files from a subset of the library of size

$K - |T|$. Then probability that a node in $B_r(u)$ caches W_j is at most $M/(K - t)$. Hence, for every $W_j \in S$,

$$\begin{aligned} \Pr[X_{u,j} = 1 | \{X_{u,j'} = x_{u,j'}, W_{j'} \in S \setminus \{W_j\}\}] \\ \geq \left(1 - \frac{M}{K - \sum_{W_{j'} \in S \setminus \{W_j\}} x_{u,j'}}\right)^{2r(r+1)} \\ \geq \left(1 - \frac{M}{K - M + 1}\right)^{2r(r+1)} = e^{-\Omega(1)} = p, \end{aligned}$$

where $|B_r(u) \setminus \{u\}| = 2r(r+1) = \Theta(K/M)$ and hence p is a constant. Let $Z = \sum_{W_j \in S} X_{u,j}$ and then $\mathbb{E}[Z] \geq s \cdot p$. Using a Chernoff bound for moderately correlated indicator random variables (e.g., see the Appendix) implies that

$$\Pr[Z < sp/2] = o(1/n^2).$$

Therefore $B_r(u)$ does not contain any replica of at least $p/2$ fraction of files cached at u , namely S' . Using the union bound over all nodes we deduce the similar statement for every node with probability at least $1 - o(1/n)$. Therefore, for every u we have,

$$\Pr[u \text{ severs a request}] \geq \frac{|B_{r/2}(u)|}{n} \cdot \frac{|S'|}{K} = \Omega(1/n)$$

where it follows from $|S'| = \Theta(M)$, $|B_{r/2}(u)| = \Theta(K/M)$. Since there are n requests, it is easy to see that the load of each server is bounded from below by a Poisson distribution $\text{Po}(c)$, where c is a constant. On the other hand, it is known that (e.g., see [33]) the maximum number taken by n Poisson distribution $\text{Po}(c)$ is $\Omega(\log n / \log \log n)$ w.h.p. and hence the lower bound is proved. \square

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

$$C = \begin{cases} \Theta(\sqrt{K/M}) & : 0 < \gamma < 1, \\ \Theta(\sqrt{K/M \log K}) & : \gamma = 1, \\ \Theta(K^{1-\gamma/2}/\sqrt{M}) & : 1 < \gamma < 2, \\ \Theta(\log K/\sqrt{M}) & : \gamma = 2, \\ \Theta(1/\sqrt{M}) & : \gamma > 2. \end{cases} \quad (7)$$

Proof of Theorem 3. Assume a request from an arbitrary node u for file W_j . The probability that this file is cached at another node v is $q_j := 1 - (1 - p_j)^M$. Cache content placement at different nodes is independent. Thus, the number of nodes which should be probed is a geometric random variable with success probability q_j . This results in the average $1/q_j$ trials that leads to expected distance of

$$\Theta\left(\frac{1}{\sqrt{q_j}}\right) = \Theta\left(\frac{1}{\sqrt{1 - (1 - p_j)^M}}\right). \quad (8)$$

When averaged over different files we will have

$$C = \sum_{j=1}^K p_j \Theta \left(\frac{1}{\sqrt{1 - (1 - p_j)^M}} \right). \quad (9)$$

- For Uniform distribution we have $p_j = 1/K$ and then

$$C = \Theta(\sqrt{K/M}). \quad (10)$$

- For Zipf distribution with $M = \Theta(1)$ we have

$$\begin{aligned} C &= \sum_{j=1}^K p_j \Theta \left(\frac{1}{\sqrt{1 - (1 - p_j)^M}} \right) \\ &= \sum_{j=1}^K p_j \Theta \left(\frac{1}{\sqrt{p_j M}} \right) \\ &= \Theta \left(\frac{\sum_{j=1}^K j^{-\gamma/2}}{\left(M \sum_{j=1}^K j^{-\gamma} \right)^{1/2}} \right). \end{aligned} \quad (11)$$

Define $\Lambda(\gamma) := \sum_{j=1}^K j^{-\gamma}$, for every γ . On the other hand it is known that for every $\gamma > 0$ (e.g., see [34])

$$\Lambda(\gamma) = \begin{cases} \Theta(K^{1-\gamma}), & 0 < \gamma < 1, \\ \Theta(\log K), & \gamma = 1, \\ \Theta(1), & \gamma > 1. \end{cases} \quad (12)$$

Now inserting the above equations into (11) completes the proof. \square

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. We address these questions in the following section.

IV. PROXIMITY-AWARE TWO CHOICES STRATEGY

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

Definition 3 (Strategy II: 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 $r = \infty^2$). *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 $i = 1, \dots, n$) will be a $\text{Po}(1)$ 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 [6]. In this model there are 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 [6]. 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. [6], 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 [6]. 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 cached inside the network. On the other hand, each file type is requested $\text{Po}(1)$ 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 [33]). 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 but this is not always the case. Example 3 shows 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 the Zipf parameter.*

Roughly speaking, when $M = 1$, we may partition the servers based on their cached file and hence we have K "dis-

²It 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.

joint" 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 $K \ll n$.

Proof of Example 3. It is easy to see that for $M = 1$, the number of caching servers with a specific file, say W_j denoted by S_j , is distributed as a $\text{Bin}(n, p_j)$. Thus applying a Chernoff bound for S_j (e.g., see the Appendix) implies that

$$\Pr[|S_j - \mathbf{E}[S_j]| \geq \mathbf{E}[S_j]/2] \leq 2\exp(-p_j n/12).$$

Moreover, let R_j denote the number of requests for file W_j , which is the sum of n i.i.d. $\text{Bin}(n, p_j)$ random variables. Again applying a Chernoff bound (e.g., see the Appendix) for Poisson random variables yields that

$$\Pr[|R_j - \mathbf{E}[R_j]| \geq \mathbf{E}[R_j]/2] \leq 2\exp(-p_j n/12).$$

Notice that $\mathbf{E}[S_j] = \mathbf{E}[R_j] = np_j$. Suppose that \mathcal{A}_j denotes the event that $|S_j - \mathbf{E}[S_j]| \leq \mathbf{E}[S_j]/2$ and $|R_j - \mathbf{E}[R_j]| \leq \mathbf{E}[R_j]/2$. Then we have that $\Pr[\mathcal{A}_j] \geq 1 - 4\exp(-p_j n/12)$. Also define \mathcal{E}_j to be the event that the two-choice model with S_j bins (caching servers) and R_j balls (requests) achieves maximum load $R_j/S_j + \Theta(\log \log S_j)$. It is shown that this event happens with probability $1 - O(1/S_j^c)$, for every constant c (e.g., see [6]). So we have that

$$\begin{aligned} \Pr[\mathcal{E}_j] &= \Pr[\mathcal{E}_j | \mathcal{A}_j] \Pr[\mathcal{A}_j] + \Pr[\mathcal{E}_j | \neg \mathcal{A}_j] \Pr[\neg \mathcal{A}_j] \\ &> (1 - 2(p_j n)^{-c})(1 - 4\exp(-p_j n/12)) \\ &\quad + (1 - 2(p_j n)^{-c})(4\exp(-p_j n/12)) \\ &\geq 1 - 8(p_j n)^{-c}. \end{aligned}$$

Since we have K disjoint subsystems, the union bound over all subsystems shows that the two choice model does achieve the desired maximum load with probability $1 - 8 \sum_{j=1}^K (np_j)^{-c} = 1 - o(1)$ which concludes the proof due to example's assumption on popularity profile.

Now we show that the Uniform and Zipf distributions satisfy the example's assumption. When \mathcal{P} is the Uniform distribution over K files, $\forall j, p_j \cdot n = n^\epsilon$. Now by setting $c = 3/\epsilon$, we have that

$$\sum_{j=1}^K (np_j)^{-c} = K(1/n^\epsilon)^c = K/n^3 = o(1/n^2).$$

Also, for Zipf distribution we have

$$p_j = \frac{j^{-\gamma}}{\sum_{j=1}^K j^{-\gamma}} = \frac{j^{-\gamma}}{\Lambda(\gamma)}.$$

Depending on γ , we consider two cases:

- $\gamma \geq 1$: For every $c > 1$ we have

$$\left(\frac{\Lambda(\gamma)}{n}\right)^c \Lambda(\gamma c) = \Theta\left(\frac{\log^c K}{n^c}\right) \Lambda(\gamma c) = o(1),$$

where we used $K < n$ and (12).

- $0 < \beta < 1$: By setting $c = 2/\gamma$ and using the fact that $K < n$, we have

$$\begin{aligned} \left(\frac{\Lambda(\gamma)}{n}\right)^c \Lambda(\gamma c) &= \Theta\left(\frac{K^{(1-\gamma)c}}{n^c}\right) \leq n^{(1-\gamma)c-c} \Lambda(\gamma c) \\ &= n^{-\gamma c} \Lambda(\gamma c) = o(1), \end{aligned}$$

where we applied (12). □

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 $C = \Theta(\sqrt{n})$. 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 the 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 Strategy 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 Strategy 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 $0 < \alpha, \beta < 1/2$ be 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.

Proof Sketch. In order to prove this theorem, due to the cache size limitation and proximity constraint (which are characterized by the parameters α and β , respectively), the classic balls-into-bins results are not applicable. The main idea behind our proof is reducing the problem to the problem of balls-into-bins on graphs. Suppose instead of choosing two random bins one chooses bins as nodes of a randomly chosen edge from a graph. Then, simply put, the result in [28] (formally stated as Theorem 5 below) says that if the graph is dense-enough then the power of two choices is achievable. In our distributed cache network setup, we prove that the proposed server selection process is equivalent to choosing the bins on a *Configuration Graph*. Moreover, we show that if $\alpha + 2\beta \geq 1 + 2 \log \log n / \log n$ then the configuration graph is dense and thus according to the result in [28] the power of two choices is achievable.

Remark 4. 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 [28] as follows.

Theorem 5 ([28]). Given an almost Δ -regular graph³ 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 5. Note that in the original theorem presented in [28], 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 the 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.

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

$$\Pr[t(u) = s] = \frac{\binom{K}{s} \binom{M-1}{M-s}}{\binom{K+M-1}{M}}, \quad (13)$$

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

In order to bound (13), 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

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

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

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

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

$$\Pr[\exists u \in V : t(u) \leq \delta M] = n^{-\omega(1)}. \quad (14)$$

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

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

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

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

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

$$\Pr[t(u, v) \geq \mu] = O(1/n^3). \quad (15)$$

³A graph is said to be almost Δ -regular, if each vertex has degree $\Theta(\Delta)$.

Hence, $t(u, v) < \mu$ w.h.p. Putting (14) and (15) together concludes the proof. \square

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

Lemma 3. *Conditioning on “goodness” of the 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 an arbitrary node u with s distinct files. Then by definition of H , for every node v we have

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

where $1 \leq s \leq M$. On the other hand u and v are connected in H , if in addition we have $d_G(u, v) \leq 2r$. Therefore for every given node u with s distinct cached files, $d(u)$ in H (degree of u in H) has a binomial distribution $\text{Bin}(b_{2r}(u), p_s)$, where $b_{2r}(u) = |B_{2r}(u)|$. Hence applying a Chernoff bound implies that with probability $1 - n^{-\omega(1)}$, we have

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

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

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

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

$$e(H) = \Theta(nM^2 r^2 / K) = \Theta(M^2 r^2), \quad (16)$$

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 $\text{Bin}(b_r(u), p)$, where $b_r(u) = |B_r(u)|$. So

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

where $b_r(u) = \Theta(r^2)$ for every u . Since $\alpha + 2\beta \geq 1 + 2 \log \log n / \log n$, we have $\mathbf{E}[F_j(u)] = \omega(\log n)$, 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

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

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,

$$\begin{aligned} \Pr[(u, v) \in E(H) \text{ is picked by Strategy II} | t(u, v) = t] \\ &= \sum_{j \in T(u, v)} \frac{1}{K} \sum_{w \in S_{u,v}} \frac{1}{n} \frac{1}{\binom{F_j(w)}{2}} \\ &= \frac{1}{n^2} \sum_{j \in T(u, v)} \sum_{w \in S_{u,v}} \frac{1}{\binom{F_j(w)}{2}} \\ &= \frac{1}{n^2} \sum_{j \in T(u, v)} \sum_{w \in S_{u,v}} \Theta(n^2 / r^4 M^2). \end{aligned} \quad (17)$$

Conditioned on “goodness,” we have for every $(u, v) \in E(H)$, $1 \leq t(u, v) < \mu$. So (17) can be simplified as

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

where the last equality follows from (16). \square

Proof of Theorem 4. Applying Lemma 3 shows that w.h.p. the configuration graph H is an almost Δ -regular graph where $\Delta = M^2 r^2 / n$. 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 $\Delta = M^2 r^2 / n = n^{2\alpha+2\beta-1} > n^\alpha$. \square

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 $C = \Theta(n^\beta)$ 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 ([28]) 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 $C = \Theta(r) = \Theta(n^\beta)$. \square

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

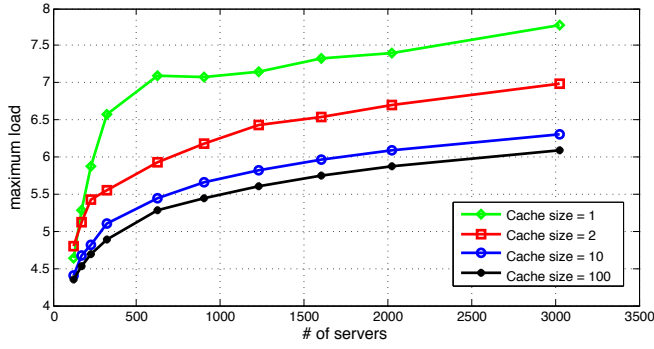


Fig. 3: The maximum load versus number of servers for Strategy I. Here, the network topology is a torus, the file popularity is Uniform, and we have $K = 100$.

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* strategies. Our simulations are implemented in Python where the code is available online at [35].

Figure 3 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 is in agreement with the logarithmic growth of the maximum load, asymptotically proved in Section III, even for the intermediate values of $n \approx 100, \dots, 3000$, which makes the results of Section III 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.

Furthermore, Figure 4 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 5 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. Figure 5 demonstrates the system performance for large system sizes, i.e., $n \approx 10^4, \dots, 10^5$.

It should be noted that in Figure 5, for the cases of $M = 1$ and $M = 2$, we observe peaks in the curves of maximum load

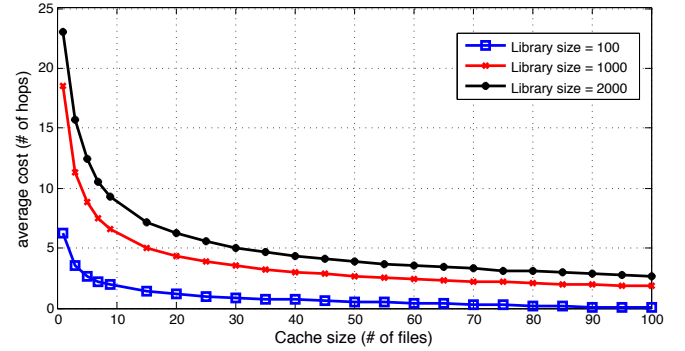


Fig. 4: The communication cost versus cache size for Strategy I. Here, the network topology is a torus of size 2025 and the file popularity is Uniform.

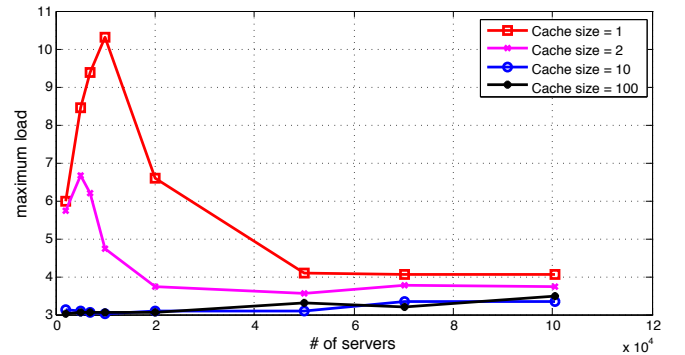


Fig. 5: The maximum load versus number of servers for Strategy II. Here, the network topology is torus, the file popularity is Uniform, and the library size is $K = 2000$. Moreover, we assume $r = \infty$.

versus number of servers. The reason behind this phenomenon is that for small number of servers Strategy II cannot achieve the power of two choices, due to low file replication inside the network. However, for large number of servers, which leads to high file replication, we have a significant drop in the maximum load due to the availability of the power of two choices. Different behaviour of the maximum load in these two operating regimes is the reason behind observing the peaks. Also, to get a better understanding of this phenomenon, let us compare the load balancing performance of Strategies I and II in Figure 6 where the file library size is $K = 100$ and $n \approx 10^3$.

In Figure 6, 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 6; for example in the curve corresponding to $M = 1$ for $n \leq 400$ we have a fast growth in maximum load which mimics the load balancing performance of Strategy I. We see the same trend for the curve corresponding to $M = 2$ for $n \leq 200$. However, assuming $M \geq 2$, for $n > 1000$, since there is enough file replication in the network, the load balancing performance is greatly improved due to the power of two choices. This is in accordance with the lessons learned from Section IV. Also, in

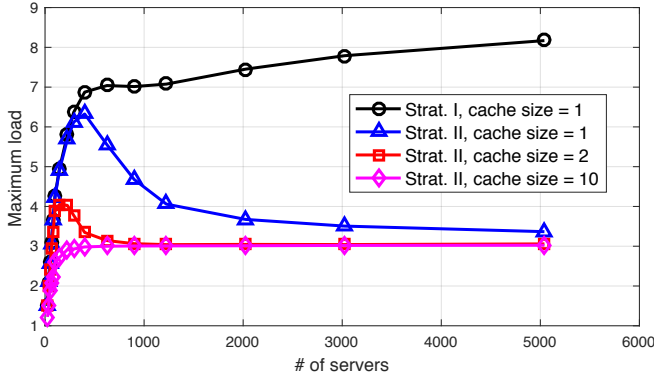


Fig. 6: The maximum load versus number of servers for Strategies I and II. Here, the network topology is torus, the file popularity is Uniform, and the library size is $K = 100$. Moreover, we assume $r = \infty$.

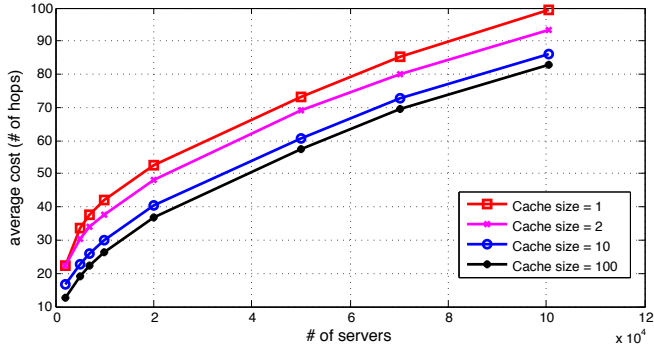


Fig. 7: The communication cost versus number of servers for Strategy II. Here, the settings are similar to that of Figure 5.

between, we have a transition region where a mixed behavior is observed. Observations made above from Figure 6 have 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 7 draws the communication cost versus number of servers for various cache sizes for similar setting used in Figure 5. Since in this figure there is no constraint on the proximity, the communication cost growth is of order $\Theta(\sqrt{n})$.

In simulations presented so far, we only considered the case $r = \infty$. In order to investigate the effect of parameter r on the performance of the system, in Figure 8, 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 8. 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 of 5000 simulation runs.

In this figure, like before (i.e., Figure 6), we observe two performance regimes based on the file replication in the network. 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.

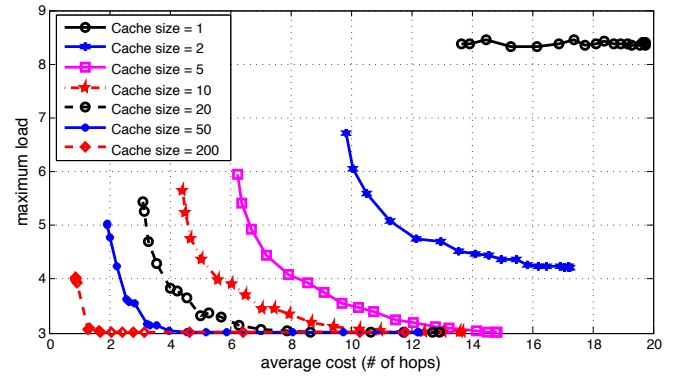


Fig. 8: The trade-off between the maximum load and communication cost for Strategy II. Here, the network topology is a torus of size 2025, the file popularity is Uniform, and the library size is $K = 500$.

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.

All above simulations investigated the performance for networks with torus topology and Uniform file popularity distribution, being in agreement with our theoretical results' indications. However, one may ask how sensitive are our findings to the network topology and file popularity choices. Thus, in the following, we examine network performance for Zipf file popularity profile and more practical network topologies, namely, random geometric graph (RGG) and power law random graph model [36].

Figures 9, 10, and 11, show the maximum load versus the number of servers for different network topologies (namely, torus, RGG, and power law model) and Zipf distribution with parameter⁴ $\gamma \in \{0, 1, 1.5\}$.

Also Figures 12, 13, and 14, demonstrate the performance trade-off between maximum load and communication cost for different network topologies (namely, torus, random RGG, and power law model) and Zipf distribution with parameter $\gamma \in \{0, 1, 1.5, 2\}$. All these simulations show that the trends and the trade-off we found in our theoretical results are also valid for more practical network settings.

It should be noted that since simulation parameters are varying in different figures, we have used smaller networks where simulation is more time-consuming (e.g., we have used smaller networks in random graphs compared to the 2D-grid, due to more time consuming shortest path finding algorithms). For convenience, a summary of simulation parameters are stated in Table I.

VI. DISCUSSION AND CONCLUDING REMARKS

In this section, we first discuss a number of important practical issues related to our proposed scheme, then we will conclude the paper.

⁴Note that the Zipf distribution with $\gamma = 0$ corresponds to the Uniform distribution.

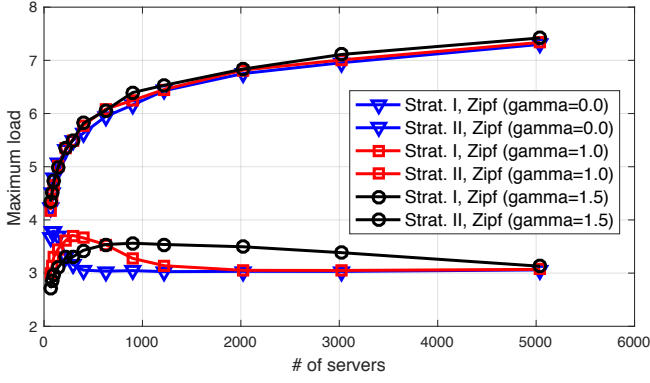


Fig. 9: The maximum load versus number of servers for Strategies I and II. Here, the network topology is torus, the file popularity is Zipf, the library size is $K = 64$, and the cache size is $M = 2$. Moreover, we assume $r = \infty$. Each point is an average over 1000 independent simulation runs.

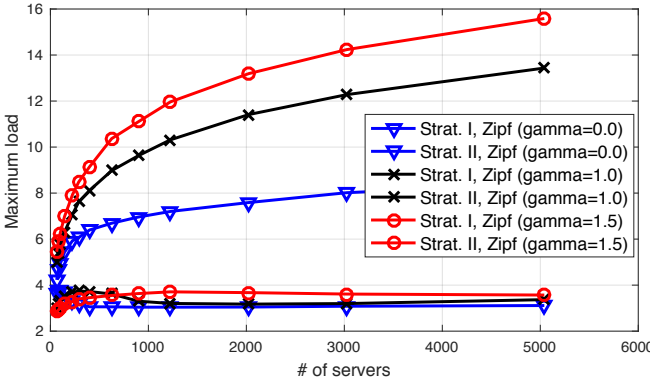


Fig. 10: The maximum load versus number of servers for Strategies I and II. Here, the network topology is RGG. The remaining settings are similar to that of Figure 9.

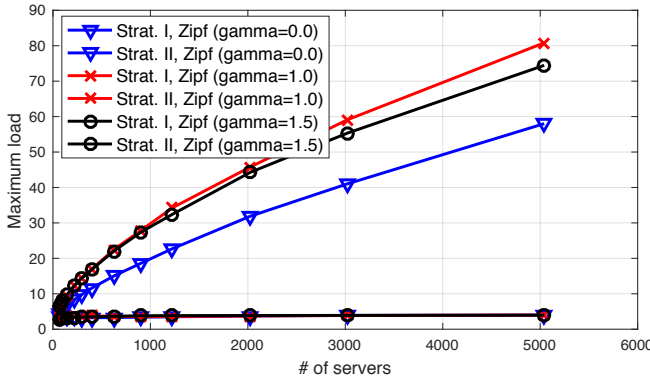


Fig. 11: The maximum load versus number of servers for Strategies I and II. Here, the network topology is power law random graph. The remaining settings are similar to that of Figure 9.

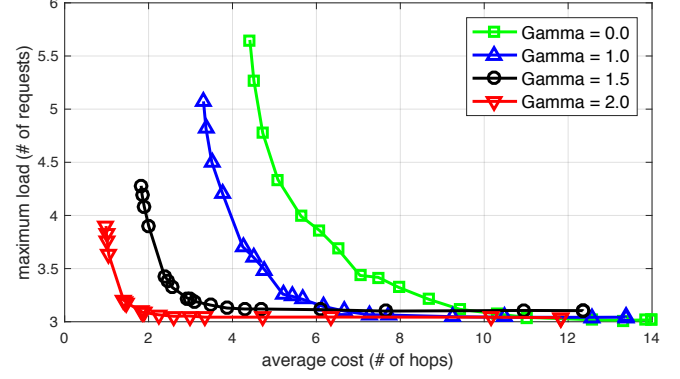


Fig. 12: The trade-off between the maximum load and communication cost for Strategy II. Here, the network topology is a torus of size 2025, the file popularity is Zipf, the library size is $K = 500$, and the cache size is $M = 10$. Each point is an average over 5000 independent simulation runs.

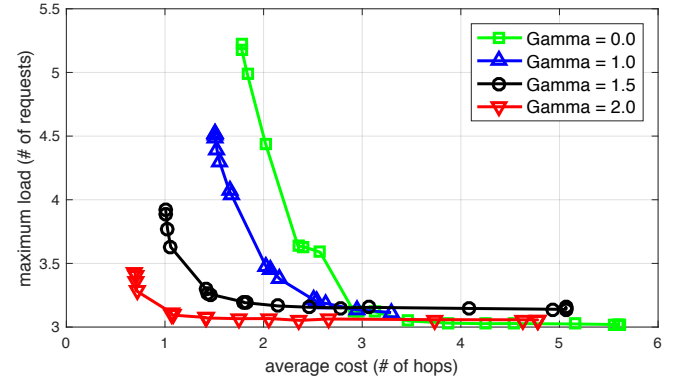


Fig. 13: The trade-off between the maximum load and communication cost for Strategy II. Here, the network topology is a RGG of size 2025. The remaining settings are similar to that of Figure 12.

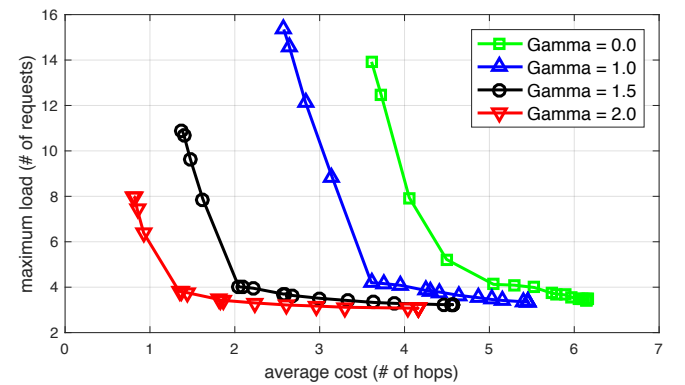


Fig. 14: The trade-off between the maximum load and communication cost for Strategy II. Here, the network topology is a power law random graph of size 2025. The remaining settings are similar to that of Figure 12.

Fig. Number	Net. Topology	Strategy	Popularity
3, 4	Torus	Strategy I	Uniform
5, 7	Torus	Strategy II ($r = \infty$)	Uniform
6	Torus	Strategy I, II ($r = \infty$)	Uniform
8	Torus	Strategy II (r varying)	Uniform
9	Torus	Strategy I, II ($r = \infty$)	Zipf
10	RGG	Strategy I, II ($r = \infty$)	Zipf
11	Power Law	Strategy I, II ($r = \infty$)	Zipf
12	Torus	Strategy II (r varying)	Zipf
13	RGG	Strategy II (r varying)	Zipf
14	Power Law	Strategy II (r varying)	Zipf

TABLE I: Summary of the simulation parameters for each figure.

Our theoretical results in Sections III and IV are stated for a 2D-Grid topology. The main reason for assuming this rather unrealistic topology is developing the main idea of the paper clearly. However, it should be noticed that our approach can be extended to more general graph models at the expense of lengthy proofs and calculations. For example, as mentioned in Section IV, $|B_r(u)|$ is the size of the ball of radius r around node u . The main feature of the 2D-Grid which affects our results is that $|B_r(u)| = \Theta(r^2)$ for all u . Now suppose that, instead of assuming a 2D-Grid, we consider a graph in which $|B_r(u)| = \Theta(r^{\dim})$, w.h.p. Then the parameter \dim will appear in our results instead of $\dim = 2$ in the special case of 2D-Grid. For example, the term $\alpha + 2\beta$ in the statement of Theorem 4 would be generalized to $\alpha + \dim \times \beta$. More generally, even for $|B_r(u)| = \Theta(f(r))$, our basic technicalities can be extended too. However, in this paper we investigate other network topologies, such as Random Geometric Graphs and Scale-Free (power law) networks via extensive simulations in Section V. As we have discussed it there, the main trends are also valid for these topologies as well. Thus, our findings cover a wider class of graph models which are more similar to real-world CDN network topologies.

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 see Distributed Hash Table (DHT) schemes, e.g., [37] and [38]). 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 [7] and [39], 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 the queuing theory based model. We postpone a rigorous analysis of such scenario to future work.

In this paper, we have assumed that the number of requests and the number of servers are the same. The main reason for this assumption is that, intuitively speaking, when we have small number of requests compared to the number of servers, no server will be overloaded. On the other hand if we have large number of requests, the deviation of server load from the average load is negligible (see [10] for a discussion in the classical balls-into-bins setup). Thus, the only interesting regime for analysis is the critical regime where we have the same number of requests and servers. A rigorous analysis for other regimes is postponed to future work.

It should be noted that in this work we have considered hop count as the proximity metric. Using this metric does not consider propagation or capacity limitations inside the network. Furthermore, in the mobile content delivery scenarios, other channel characteristics should be taken into account as discussed in [40] and [41], thus our results are more indicative of the performance of wired networks. Considering other communication issues into our analytical framework such as propagation, link capacities and mobile channel specifications are interesting directions for future work.

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

ACKNOWLEDGMENT

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

REFERENCES

- [1] A. Pourmiri, M. Jafari Siavoshani, and S. P. Shariatpanahi, "Proximity-aware balanced allocations in cache networks," in *IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, 2017.
- [2] Cisco, "Cisco visual networking index: global mobile data traffic forecast update, 2013/2018," *White Paper*, 2014.
- [3] G. Zhang, Y. Li, and T. Lin, "Caching in information centric networking: A survey," *Computer Network*, vol. 57, pp. 3128–3141, 2013.
- [4] 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.
- [5] 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.
- [6] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal, "Balanced allocations," *SIAM J. Comput.*, vol. 29, no. 1, pp. 180–200, 1999.
- [7] M. Mitzenmacher, "The power of two choices in randomized load balancing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, no. 10, pp. 1094–1104, 2001.
- [8] 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.
- [9] C. Lenzen and R. Wattenhofer, "Tight bounds for parallel randomized load balancing," *Distributed Computing*, vol. 29, no. 2, pp. 127–142, 2016.

- [10] 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.
- [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] A. Meyerson, K. Munagala, and S. Plotkin, "Web caching using access statistics," in *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '01, 2001, pp. 354–363.
- [19] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *2010 Proceedings IEEE INFOCOM*, March 2010, pp. 1–9.
- [20] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Information Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.
- [21] S. Gkitzenis, G. S. Paschos, and L. Tassioulas, "Asymptotic laws for joint content replication and delivery in wireless networks," *IEEE Transactions on Information Theory*, vol. 59, no. 5, pp. 2760–2776, May 2013.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] K. Kenthapadi and R. Panigrahy, "Balanced allocation on graphs," in *Proc. 17th Symp. Discrete Algorithms (SODA)*, 2006, pp. 434–443.
- [29] B. Godfrey, "Balls and bins with structure: balanced allocations on hypergraphs," in *Proc. 19th Symp. Discrete Algorithms (SODA)*, 2008, pp. 511–517.
- [30] A. Pourmirmir, "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.
- [31] 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.
- [32] 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.
- [33] 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.
- [34] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," *CoRR*, vol. abs/1502.03124, 2015. [Online]. Available: <http://arxiv.org/abs/1502.03124>
- [35] [Online]. Available: https://github.com/mahdijafaris/Balls_and_Bins
- [36] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.
- [37] 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.
- [38] 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.
- [39] 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.
- [40] A. S. Cacciapuoti, M. Caleffi, M. Ji, J. Llorca, and A. M. Tulino, "Speeding up future video distribution via channel-aware caching-aided coded multicast," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2207–2218, Aug 2016.
- [41] M. Ji and R. R. Chen, "Caching and coded multicasting in slow fading environment," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, March 2017, pp. 1–6.



Mahdi Jafari Siavoshani is an Assistant Professor in the Department of Computer Engineering, Sharif University of Technology (SUT), Tehran, Iran, from 2013. He received his B.Sc. degrees in Communication Systems as well as Physics both from SUT in 2005. He was awarded an Excellency scholarship from Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, where he received the M.Sc. degree in 2007 and the Ph.D. degree in 2012, both in Computer, Communication, and Information Sciences. After completing his Ph.D., he joined the Institute of Network Coding (INC) at the Chinese University of Hong Kong (CUHK) as a postdoctoral fellow from 2012 to 2013. His research interests include finding fundamental limits of Communication Systems and Networks. Moreover, he is interested in the connection between Communication and Computer Sciences. In particular, he is interested in theory and algorithms for network information flow, network coding, network algorithms, information and coding theory, wireless communication networks.



Ali Pourmirmir is an assistant professor at CE department of University of Isfahan. Prior to joining UI, he was a postdoctoral fellow in CS school of IPM and he got a PhD from Max-Planck-Institute for Informatics in 2015. He received his M.Sc. (2007) and B.Sc. (2004) in Applied Mathematics from Isfahan University of Technology and University of Isfahan, respectively. Broadly speaking, he is interested in design and analysis of (randomized) algorithms, more specifically problems arising in networks, such as information diffusion, random walks and load balancing on networks.



Seyed Pooya Shariatpanahi received the B.Sc., M.Sc., and Ph.D. degrees from the Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran, in 2006, 2008, and 2013, respectively. He is currently a senior researcher with the Institute for Research in Fundamental Sciences (IPM), Tehran, Iran. His research interests include wireless networks, information theory, and complex systems. He was a recipient of the Gold Medal at the National Physics Olympiad in 2001.