Bipartite graph structures for efficient balancing of heterogeneous loads

Mathieu Leconte, Lelarge Marc, Massoulié Laurent

Bipartite Graph Structures for Efficient Balancing of Heterogeneous Loads

M. Leconte
Technicolor - INRIA
mathieu.leconte@inria.fr

M. Lelarge
INRIA - École Normale Supérieure
marc.lelarge@ens.fr

L. Massoulié
Technicolor
laurent.massoulie@technicolor.com

ABSTRACT

This paper considers large scale distributed content service platforms, such as peer-to-peer video-on-demand systems. Such systems feature two basic resources, namely storage and bandwidth. Their efficiency critically depends on two factors: (i) content replication within servers, and (ii) how incoming service requests are matched to servers holding requested content. To inform the corresponding design choices, we make the following contributions.

We first show that, for underloaded systems, so-called *proportional content placement* with a simple greedy strategy for matching requests to servers ensures full system efficiency provided storage size grows logarithmically with the system size. However, for constant storage size, this strategy undergoes a phase transition with severe loss of efficiency as system load approaches criticality.

To better understand the role of the matching strategy in this performance degradation, we characterize the asymptotic system efficiency under an *optimal* matching policy. Our analysis shows that –in contrast to greedy matching– optimal matching incurs an inefficiency that is exponentially small in the server storage size, even at critical system loads. It further allows a characterization of content replication policies that minimize the inefficiency. These optimal policies, which differ markedly from *proportional placement*, have a simple structure which makes them implementable in practice.

On the methodological side, our analysis of matching performance uses the theory of local weak limits of random graphs, and highlights a novel characterization of matching numbers in bipartite graphs, which may both be of independent interest.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems

SIGMETRICS'12, June 11–15, 2012, London, England, UK.

General Terms

Theory, Performance

Keywords

Content Delivery Networks, Random Graphs, Matchings, Content Placement Policies

1. INTRODUCTION

The surge in consumption of video over the Internet necessitates massive bandwidth provisioning. At the same time, storage is extremely cheap. Extensive replication of content not only within data centers, but also at the periphery of the network, e.g. in users' computers, can thus be envisioned to leverage uplink bandwidth available from users' homes. *People's CDN* is one particular commercial initiative in this direction; the massively popular *PPLive* peer-to-peer system for video-on-demand is another example of this approach.

Such systems feature two key resources, namely storage and bandwidth. Ideally, one would like to utilize storage by pre-loading content replicas at individual servers, in such a way that bandwidth of all servers is available to serve any incoming request. In other words, a challenge in engineering such systems is to create content replicas so that bandwidth can be used maximally. Several strategies for content replication have been considered: uniform replication does not discriminate between contents; proportional replication tunes the number of replicas to the average number of requests, and automatically arises at cache memories when the so-called random-useful cache eviction method is used (it is also approximately achieved by the classical least-recently-used eviction rule [21]).

Our first objective is to develop a clear understanding of the relative merits of distinct replication strategies in the context of large-scale distributed server platforms. We also aim at characterizing the amount of storage necessary to remove the *content bottleneck*, i.e. for what amount of storage is the system service capacity only constrained by its overall bandwidth? These properties certainly depend both on the content replication strategy and on the algorithm used to match incoming service requests to servers capable of treating these requests.

To this end, we develop performance models based on bipartite graphs (Section 2). To motivate our main results, in Section 3 we first consider simple strategies for these two design issues, namely proportional replication and greedy online matching. Regarding proportional replication, using simple large-deviations estimates (Chernoff bounds) we

show that a limited amount of storage (logarithmic in system size) suffices to absorb all requests in *sub-critical scenarios*, where the system is under-loaded. In other words, the content bottleneck is removed under these assumptions. However, these analysis techniques are not powerful enough to characterize system performance under critical or super-critical load. We also establish through mean field analysis that system inefficiency is exponentially small in the amount of storage per server, provided the system is not critically loaded. In contrast, the performance degrades significantly at critical loads.

We thus turn to other analysis tools to bound the best possible inefficiency under general replication strategies and system loads. In Section 4, we exploit recent advances [17, 12] in the application of the *cavity method* from statistical physics to obtain explicit characterizations of the optimal system capacity under given replication strategies (Theorem 2). We also establish a novel characterization of the matching numbers in finite bipartite graphs (Theorem 1) which helps explain the formulas of Theorem 2. In addition, this result suggests that the applicability of the cavity method could be extended to broad classes of bipartite random graph models beyond locally tree-like graphs, as it was recently observed in related frameworks by [7] and [5].

We then leverage these tools to revisit our initial questions in Section 5. There, we establish that system inefficiency under optimal matching is exponentially small in storage space, and this even at criticality, for a large class of replication policies which includes uniform and proportional. We furthermore identify the best replication policies; they have a simple distinct structure for each of the three possible system regimes, namely sub-critical, critical and super-critical. We review related work in Section 6 and conclude in Section 7.

2. SYSTEM MODEL AND STATISTICAL ASSUMPTIONS

In this section we introduce the bipartite graph representation of the system on which our analysis is based. We also describe some statistical assumptions on the contents and the replication policies at the servers, as well as the random graph models that they induce.

2.1 Content-server graph

We are given a collection of n contents and m servers. Each server is storing replicas of a subset of the contents. We assume that all servers have an identical storage capacity $d \in \mathbb{N}$, i.e. each server is capable of storing d different contents at then same time. Such a system can be represented by a bipartite graph $G = (C \cup S, E)$ where:

- the set C represents the contents and is of cardinality n;
- the set S represents the servers and is of cardinality m;
- there is an (undirected) edge $(cs) \in E$ whenever server s stores a replica of content c.

We denote by ∂c or ∂s the neighborhood of c or s, i.e. the set of neighbors in G of c or s. Hence, ∂c is the set of servers that hold a replica of content c, and ∂s is the set of contents of which server s stores a replica. The degree of a node $s \in S$ is exactly $d = |\partial s|$, while the degree of a node $c \in C$ is equal to the total number of replicas of content c in the system. In Section 2.2, we will explain how the graph G is constructed. Before, we describe how the natural constraints on the service of requests can be seen on the graph G.

We let $b_c \in \mathbb{N}$ be the number of requests made for content $c \in C$. Such a request for content c can be served only by a server having content c stored on disk, i.e. a request for content c can only be served by a neighbor of c in the bipartite graph G. Moreover, we assume in this paper that servers can serve at most one request at a time, although an analysis could be attempted using similar methods when servers have a larger service capacity. Hence, if a request for a content c is assigned to server $s \in \partial c$, then server s becomes unavailable for all the contents in $\partial s \backslash c$, where $\partial s \backslash c$ is a short-hand notation for $\partial s \setminus \{c\}$.

We are now ready to define an allocation which represents a possible assignement of requests to servers. Given a bipartite graph $G = (C \cup S, E)$, and a vector $B = (b_c)_{c \in C}$, an allocation is a subset of edges $A \subseteq E$ such that

- each server in S is adjacent to at most one edge in A, as a server is able to serve at most one request.
- each content c in C is adjacent to at most b_c edges in A: it would make no sense to dedicate more servers to the service of content c than there are requests for c.

The total number of requests satisfied by an allocation A is simply given by the cardinality $|A|$ of the set A since if edge $(cs) \in A$ one request for content c is served by server s. The maximum size of such an allocation for a given graph G is denoted by $M(G, B)$. In the particular case where $b_c = 1$ for all contents $c \in C$, an allocation of G is what is classically known as a *matching*, and $M(G, B) = M(G)$ is then the *matching number* of G.

We define the load of the system as ratio of the total number of requests summed over all contents and the total service capacity summed over all servers, thus $\rho = \sum_c b_c/m$. If $\rho < 1$ (resp. $\rho > 1$, $\rho = 1$), we say that the system is under-loaded (resp. over-loaded, critical). Clearly, in the under-loaded regime not all servers can be assigned to a request, while in the over-loaded regime not all requests can be serviced. Hence, we have $|A| \leq \min\left(\sum_{c \in C} b_c, m\right)$ and we define the inefficiency of an allocation A as the quantity $\gamma_A = \rho \wedge 1 - |A|/m \geq 0$, where $\rho \wedge 1 = \min(\rho, 1)$. In the under-loaded regime, the inefficiency represents the ratio of the total number of unallocated requests divided by the total number of servers. In the over-loaded regime, the inefficiency represents the fraction of unallocated servers.

The objective is to minimize the inefficiency of the allocations output by the matching algorithm on the graph determined by the replication policy. This could result in a joint optimization problem of both the matching scheme and the replication strategy, however for tractability reasons we will consider the two problems separately. This leads to two design questions, namely: (i) find a replication policy that minimizes the minimal inefficiency over the resulting graph G, given by $\gamma_{G,B} = \rho \wedge 1 - M(G, B)/m$; (ii) find a matching algorithm which is simple and whose inefficiency approaches as close as possible to $\gamma_{G,B}$ for graphs G (and B) of interest. We will deal with both questions in the sequel.

Note that the problem of finding a maximum allocation in a finite bipartite graph is known to be of strongly polynomial complexity [4], using a strongly polynomial algorithm

for maximum flow problems by [20]. There may nonetheless exist lower-complexity methods, that yield comparable performance for the graphs of interest.

2.2 Random graph models

The large scale of the system as well as the uncertainty ahead of time in the number of requests for each content make it hopeless to design with too much precision the joint constitution of the caches of all the servers, i.e. deciding jointly of each and every edges in the bipartite graph G. Indeed, we will be interested in a regime where the number of contents n and the number of servers m tend to infinity together, with $\frac{m}{n} \to \beta \in \mathbb{R}_+$. In this regime, it seems plausible that there will be little fine-tuned cooperation among the servers, and they will thus essentially make independent choices for their cache based on the available statistics on the requests. Of course, allowing more cooperation between servers could only improve the performance and would arguably be even more realistic, however one would have to precisely define the extent of cooperation that is reasonable; we will not deal with such issues in this paper. We will also assume that the storage size d of each server does not scale with n. Indeed the point of having a large number of servers is precisely that the storage size of each server does not need to scale with the size of the system.

We now present the most simple model we will use in this paper, which represents a situation where no information on the requests is available to the servers: in this case, each server picks d contents uniformly at random among all contents and independently of the other servers. Then, the distribution of the number of replicas of a given content, i.e. its degree in the graph G is a Binomial random variable with parameters m and $\frac{d}{n}$. Hence in the regime where n tends to infinity with $m/n \to \beta$ while d is kept fixed, the distribution of the degree of a content in G tends to a Poisson distribution with mean $d\beta$. Note that even if the degrees of the contents are not independent for any fixed n, the degrees become k-wise independent for any $k = \mathrm{o}(n)$ in the limit when n tends to infinity [6]. We will refer to this model as the *single class model*. In Section 3, we will moreover use a Poisson distribution of parameter λ for the number of requests in order to analyze the performance of a simple greedy algorithm for matching requests to servers.

Of course, in a real system, the contents are not all of the same popularity, and we may consider prioritizing replication of more popular contents at the servers. In such a system the number of requests for a content and its number of replicas would not be independent as in the previous example; however the servers may not have full information ahead of time about the number of requests for each content and they may have to rely only on a coarse-grained estimate of the popularity of the contents.

We model this situation by partitioning the set of all contents into *popularity classes*. Specifically, we let class i contain a fraction α_i of all contents and we assume that the number of requests for the contents in class i are independent and identically distributed with a Poisson distribution with mean λ_i. Servers do not distinguish between contents within the same popularity class, but they may favor some classes with respect to others. For example, we can let servers assign independently contents to each memory slot in the following way: the server first chooses a class of content, for example picking class i with probability θ_i; then it picks a particular content within that class uniformly at random. Then, the number of replicas and number of requests for a content are independent given its class. Moreover in the large n limit, they are distributed as two independent Poisson random variables: the number of replicas having mean $\beta d\theta_i/\alpha_i =: \beta_i d$ and the number of requests having mean λ_i. Of course, other types of content popularity distributions could be of interest (notably Zipf-like distributions), however the model considered here is particularly suited for investigating how to adapt the replication policy to the popularity of contents, i.e. how to choose the θ_i's as a function of the α_i's and λ_i's. It will be used for that purpose in Section 5. We call this model the *multi-class model*.

3. CANDIDATE STRATEGIES

In this section we consider easily implementable candidates for the replication policy and matching algorithm. Their analysis will serve as a point of comparison to determine whether more advanced strategies are needed or not. We first consider the so-called proportional placement policy, which maintains for each content a number of replicas proportional to its popularity, and then a simple online matching algorithm that iteratively makes random assignements of requests to available servers.

3.1 Proportional placement

Assume that the numbers of requests b_c for content c are mutually independent over all $c \in C$, b_c having a Poisson distribution with parameter $\lambda_c \geq \underline{\lambda}$ for some fixed parameter $\underline{\lambda} > 0$. Based on knowledge of the λ_c's only (and not of the b_c's), the replication problem consists in determining the number of replicas d_c of each item c and their placement onto servers. A candidate strategy consists in taking the number of replicas d_c deterministic and proportional to the *expected* number of requests λ_c: $d_c \approx (\beta d)\lambda_c/\lambda$, where $\lambda = n^{-1}\sum_{c=1}^{n} \lambda_c$. The system is then sub-critical if the following stability condition holds, that the expected number of requests be smaller than the number of servers, which reads

$$\delta := \beta - \lambda > 0, \tag{1}$$

where we introduced notation δ for the stability margin $\beta - \lambda$. In this context we have the following:

PROPOSITION 1. *Assuming $\delta > 0$ and proportional placement is used, all requests can be met with high probability if the storage space per server d satisfies $d \geq \Omega(\log(n))$.*

As a corollary, this result implies that the inefficiency of proportional placement is equal to 0 a.s. for d logarithmic in the number of contents n in the system, i.e. $\mathbb{E}[\gamma_{G,B}] = 0$ for $d \geq \Omega(\log(n))$ when G is chosen according to the proportionnal placement rule (and B is drawn as explained before).

PROOF. For each content c and each of the corresponding b_c requests, we can split the request equally into d_c subrequests of size $1/d_c$. To a given server s with corresponding collection ∂s of stored items, for each item $c \in \partial s$ we associate b_c sub-requests to that particular server. Then a service of all requests will be feasible provided that for each such server $s \in \{1, \ldots, \beta n\}$ one has

$$\sum_{c \in \partial s} \frac{b_c}{d_c} \leq 1. \tag{2}$$

Indeed, this mapping of sub-requests to servers would constitute a fractional matching of requests, and an integral matching would therefore exist, by the total unimodularity of the adjacency matrix of the graph G.

Calling $\mathcal{A}_s$ the event corresponding to Condition (2) and $\mathcal{A}_s^{\mathbf{c}}$ its complement, Chernoff's inequality yields

$$\begin{aligned}\mathbb{P}(\mathcal{A}_s^{\mathbf{c}}) &\leq \exp\left(-\sup_{\theta>0}\left[\theta - \textstyle\sum_{c\in\partial s}\log \mathbb{E}e^{\theta(b_c/d_c)}\right]\right)\\ &= \exp\left(-\sup_{\theta>0}\left[\theta - \textstyle\sum_{c\in\partial s}\lambda_c\left(-1+e^{\theta/d_c}\right)\right]\right).\end{aligned}$$

Note that $d_c \geq d\beta\lambda_c/\lambda - 1 \geq \lambda_c d\left(1+\frac{\delta}{2\lambda}\right)$ for d large enough, namely for $d \geq 2\lambda/(\delta\underline{\lambda})$. Replacing d_c by this corresponding lower bound in the above expression, we obtain

$$\mathbb{P}(\mathcal{A}_s^{\mathbf{c}}) \leq \exp\left(-\sup_{\theta>0}\left[\theta - \sum_{c\in\partial s}\lambda_c\left(-1+e^{\theta'/\lambda_c}\right)\right]\right),$$

where we introduced the notation $\theta' := \theta\lambda/(d(\lambda+\delta/2\lambda))$. It is readily checked that the function $\lambda_c \to \lambda_c\left(-1+e^{\theta'/\lambda_c}\right)$ is decreasing in λ_c, and is thus upper-bounded by its value at $\underline{\lambda}$ for all $\lambda_c \geq \underline{\lambda}$. Using the fact that the cardinality of ∂s is precisely d, this then entails

$$\begin{aligned}\mathbb{P}(\mathcal{A}_s^{\mathbf{c}}) &\leq e^{-\sup_{\theta>0}\left[\theta - d\underline{\lambda}\left(-1+e^{\theta\lambda/(d\underline{\lambda}(\delta/2+\lambda))}\right)\right]}\\ &= e^{-d\underline{\lambda}h(1+\delta/(2\lambda))},\end{aligned}$$

where $h(x) := x\log(x) - x + 1$ is the Cramér transform of a unit rate Poisson random variable. It follows that

$$\mathbb{P}(\mathcal{A}_s^{\mathbf{c}}) \leq n^{-\alpha\underline{\lambda}h(1+\delta/(2\lambda))}$$

for $d \geq \alpha\log(n)$. Thus, provided

$$\alpha > \frac{1}{\underline{\lambda}h(1+\delta/(2\lambda))}, \tag{3}$$

by the union bound the probability that at least one event $\mathcal{A}_s$ fails is $\mathrm{o}(1)$. Consequently, under the stability assumption (1), for popularities lower-bounded by $\underline{\lambda} > 0$, all requests are met with high probability as $n \to \infty$ provided

$$d \geq \max(\alpha\log(n), 2\lambda/(\delta\underline{\lambda})),$$

where constant α verifies (3). □

This result indicates that a logarithmic storage size d suffices to meet all the requests, provided one uses the above proportional placement. Note also that proportional placement is robust, in the sense that this feasibility property does not depend on the particular way in which content replicas are co-located within servers so long as one does not replicate content more than once at a single server.

Nevertheless, several questions of interest remain open. In particular, the above argument does not say what happens when the stability margin δ becomes small: indeed, for small δ, the lower bound (3) on α is $\Theta(\delta^{-2})$, and it is thus not clear how this replication policy behaves in a near-critical regime $\delta \to 0$. In addition, it does not say how many requests remain unmatched under storage size restrictions, for instance for constant rather than logarithmic d. Moreover, it states the existence of a matching, but does not address whether such a matching is easy to find. The first two of these three questions will be addressed in Section 5.

3.2 Online matching algorithm

We now study the performance of a simple online algorithm for matching incoming requests to servers. For simplicity we consider the single-class model, where all contents have the same popularity, and servers choose uniformly at random which content they store.

We consider the following algorithm: pick a request at random and match it, if possible, to an unmatched server chosen uniformly at random among all unmatched servers storing the corresponding content; otherwise the request is discarded and all further requests for the same content will be rejected as well. We call such contents *depleted* contents (as opposed to *available* contents).

We adopt a temporal view of this process, assuming that a request arrives at every time step and is for a content chosen uniformly at random. As soon as a request arrives, the algorithm checks whether it can be matched to a server or not; as this does not require knowledge of the requests that will arrive at later time steps, we say this algorithm is *online*. After λn time steps, the number of requests for the contents will be independent Poisson random variables of parameter λ, as required.

Let γ be the limit of the inefficency of the online algorithm as $n \to \infty$. We then have the following:

PROPOSITION 2. *For $\rho \neq 1$, the inefficiency γ verifies*

$$\gamma \leq \exp\left(-d\beta|\rho-1|(1+\mathrm{o}(1))\right),$$

where the $\mathrm{o}(1)$ *is with respect to the storage capacity d. In the critical regime $\rho = 1$,*

$$\gamma = \frac{\log 2}{d\beta} + \mathrm{o}(1/d).$$

Thus, online matching incurs severe performance degradation at criticality, compared to the under-loaded and over-loaded regimes. This will be contrasted with the properties of optimal matching in Section 5.

The remainder of this section describes the main steps in the proof of the Proposition 2 above.

3.2.1 Mean field analysis of online matching

We let X_t^n be the number of matched servers at time t and Y_t^n be the number of contents for which at least one request has already been discarded by time t. It can be easily verified that $(X_t^n, Y_t^n)_{t\in\mathbb{N}}$ is a Markov chain. Indeed, it is enough to check that the induced subgraph on the $n-Y_t^n$ available contents and the $m-X_t^n$ unmatched servers is still distributed according to the same random graph model, i.e. every unassigned server stores exactly d available contents (because it cannot store depleted contents as those would not be depleted otherwise) and the set of contents stored by an unmatched server is chosen uniformly at random among the subsets of size d of the set of available contents. The correct distribution of number of requests for each content is ensured by the random uniform choice of a content among all contents for each incoming request.

The one-step transition probabilities of the Markov chain are as follows:

$$(X_{t+1}^n, Y_{t+1}^n) = \begin{cases} (X_t^n, Y_t^n) & \text{w.p. } \frac{Y_t^n}{n}\\ (X_t^n, Y_t^n+1) & \text{w.p. } \frac{n-Y_t^n}{n}p_{x,y}\\ (X_t^n+1, Y_t^n) & \text{w.p. } \frac{n-Y_t^n}{n}(1-p_{x,y})\end{cases}$$

where $p_{x,y}$ is the probability that no unmatched server stores a particular available content given $X_t^n = x$ and $Y_t^n = y$. We have

$$p_{x,y} = \left(\frac{\binom{n-y-1}{d}}{\binom{n-y}{d}} \right)^{m-x} \sim_{n\to\infty} e^{-d\frac{m-x}{n-y}}.$$

In the limit $n \to \infty$, we use mean field techniques to approximate the Markov chain $(X_t^n, Y_t^n)_t$ by the solution of differential equations. We define $x(t)$ and $y(t)$ as candidate approximations of $\frac{X_{tn}^n}{n}$ and $\frac{Y_{tn}^n}{n}$ respectively, by setting: $x(0) = y(0) = 0$ and $x(t), y(t)$ are given by the following differential equations:

$$\dot{x} = (1-y)\left(1 - e^{-d\frac{\beta - x}{1-y}}\right) \quad (4)$$

$$\dot{y} = (1-y)e^{-d\frac{\beta-x}{1-y}} \quad (5)$$

Classical results of Kurtz [11] then imply the following

LEMMA 1. *Almost surely, we have*

$$\lim_{n\to\infty} \frac{1}{n}(X_{\lambda n}^n, Y_{\lambda n}^n) = (x(\lambda), y(\lambda)).$$

The limiting inefficiency γ of the online algorithm is thus given by $\gamma = \rho \wedge 1 - x(\lambda)/\beta$. While the above differential equations do not admit closed form solutions, it is possible to derive explicit upper and lower bounds based on more tractable ODE's, which become accurate for large d. We defer this part of the proof to the appendix.

4. OPTIMAL MATCHING

In this section, we compute the size of a maximum allocation $M(G, B)$ and hence the inefficiency $\gamma_{G,B}$ of the graph for the models presented in Section 2.2. Our first main result shows that $M(G, B)$ can be expressed in terms of 'local contributions' for any bipartite graph G and any numbers of requests B. This expression turns out to be 'continuous' in the large n limit and converge a.s. to a deterministic quantity for random graphs drawn according to the models of Section 2.2, which allows us to compute explicitly the asymptotic for the maximum number of requests satisfied.

4.1 Finite bipartite graphs

We focus here on finite bipartite deterministic graphs. The results obtained in this subsection are meant to help understand Theorem 2 on which the analysis in Section 5 builds.

We will need to introduce some notations to state the results of this section. First-of-all, recall from Section 2.1 that $G = (C \cup S, E)$ is a finite bipartite graph and b_c is the number of requests for content $c \in C$. An allocation A is a subset of edges of G which satisfies some degree constraints at the vertices of G: in the subgraph $(C \cup S, A)$ where only edges in A are kept, each node $c \in C$ has degree at most b_c and each node $s \in S$ has degree at most 1. We define $b_s = 1$ for all $s \in S$ to ease the definition of quantities that are similar for contents and servers, so that the vector $(b_v)_{v\in C\cup S}$ represents the degree constraints for the allocation A.

The characterization of $M(G, B)$ that we will obtain here involves a message passing algorithm between the vertices of G, which can be seen to be the well-known *belief propagation* algorithm from statistical physics [15, 16]. We will not expand on the links with belief propagation as this is not the purpose of this paper and no prior knowledge of the related literature will be needed here.

As neighboring vertices in G will be sending messages to each other, and although the original graph G is not oriented, it is convenient to associate to each edge $(cs) \in E$ two oriented edges denoted by $c \to s$ and $s \to c$. We let $\overrightarrow{E}$ be the set of oriented edges of G. In what follows a vector $I \in \{0,1\}^{\overrightarrow{E}}$ will be interpreted as a set of (binary) messages $I_{x\to y} \in \{0,1\}$ along oriented edges $x \to y \in \overrightarrow{E}$. We say that $I_{x\to y}$ is the message from x to y. For a given oriented edge $x \to y$, we define the local operator which takes as arguments all incoming messages to x except the one coming from y:

$$\mathcal{P}_{x\to y}\left((I_{z\to x})_{z\in\partial x\setminus y}\right) = \mathbf{1}\left(\sum_{z\in\partial x\setminus y} I_{z\to x} < b_x\right). \quad (6)$$

By convention, if x is a leaf in the graph G, i.e. y is the only neighbor of x, we set $\mathcal{P}_{x\to y} = 1$.

If J is a vector in $\{0,1\}^{\overrightarrow{E}}$ and L is a subset of $\overrightarrow{E}$, we write J_L for the vector induced by J on $\{0,1\}^L$ and $|J_L| = \sum_{\overrightarrow{e}\in L} J_{\overrightarrow{e}}$. Also, for $x \in V$, let $\overrightarrow{\partial x}$ be the set of edges directed towards x. It will be convenient to see the local operator $\mathcal{P}_{x\to y}$ as an operator from $\{0,1\}^{\overrightarrow{E}}$ to $\{0,1\}$, although it depends only on a few component of the input. Namely, with these notations, for $I \in \{0,1\}^{\overrightarrow{E}}$, we can rewrite (6) as: $\mathcal{P}_{x\to y}(I) = \mathbf{1}\left(|I_{\overrightarrow{\partial x}}| - I_{y\to x} < b_x\right)$. It is now easy to define a global operator on the graph G, denoted $\mathcal{P}_G : \{0,1\}^{\overrightarrow{E}} \to \{0,1\}^{\overrightarrow{E}}$, which performs simultaneously the action of all the local operators $\mathcal{P}_{x\to y}$ for all x, y, so that $J = \mathcal{P}_G(I)$ is defined by: $J_{x\to y} = \mathcal{P}_{x\to y}(I)$. In words, the action of $\mathcal{P}_G$ on messages I is as follows: for each oriented edge $x \to y$, return a message 0 on this edge if the sum of the incoming messages to x from neighbors different from y is at least b_x and return a message 1 otherwise.

The global operator $\mathcal{P}_G$ was introduced in [12], where it was shown that in the special case where G is a finite tree (i.e. acyclic graph) there is a unique fixed point to the operator $\mathcal{P}_G$ (obtained by iterating it) and the size of the maximum allocation can be computed from this fixed point. Here we show that this result is indeed correct for a much larger class of graphs, namely for any bipartite graph.

THEOREM 1. *Consider a finite, bipartite graph $G = (C \cup S, E)$ with $b_c \in \mathbb{N}$ for $c \in C$. We define the function $F^S : \{0,1\}^{\overrightarrow{E}} \to \mathbb{N}$ by*

$$F^S(I) = \sum_{s\in S} \mathbf{1}(|I_{\overrightarrow{\partial s}}| > 0) + \sum_{c\in C} b_c \mathbf{1}\left(\sum_{s\in\partial c} \mathcal{P}_{s\to c}(I) > b_c\right).$$

Then, we have

$$M(G, B) = \inf_{I=\mathcal{P}_G\circ\mathcal{P}_G(I)} F^S(I). \quad (7)$$

We use the superscript S for the function F^S to emphasize the fact that F^S actually depends only on messages incoming to nodes in S. Indeed, if we denote by $I_{C\to S}$ those messages, applying once the operator $\mathcal{P}_G$ to these messages give a set of messages from nodes in S to nodes in C that we denote with a slight abuse of notation: $J_{S\to C} = \mathcal{P}_G(I_{C\to S})$. Now if I is a solution of the fixed point equation $I = \mathcal{P}_G \circ \mathcal{P}_G(I)$, we must also have $\mathcal{P}_G(J_{S\to C}) = I_{C\to S}$. For such

an I, if we consider the messages $K \in \{0,1\}^{\vec{E}}$ defined by: $K_{C\to S} = I_{C\to S}$ and $K_{S\to C} = J_{S\to C}$, we have $K = \mathcal{P}_G(K)$. Moreover since $F^S(K)$ depends only on $K_{C\to S}$, we have $F^S(K) = F^S(I)$. Hence we proved that

$$M(G,B) = \inf_{I=\mathcal{P}_G(I)} F^S(I). \tag{8}$$

But if $I = \mathcal{P}_G(I)$, the expression of $F^S(I)$ simplifies to

$$F^S(I) = \sum_{s\in S} \mathbf{1}\left(|I_{\overrightarrow{\partial s}}| > 0\right) + \sum_{c\in C} b_c \mathbf{1}\left(|I_{\overrightarrow{\partial c}}| > b_c\right).$$

Now if the graph G is a finite tree, it is easy to see that there is indeed a unique solution to the fixed point equation $I = \mathcal{P}_G(I)$, so that our Theorem 1 recovers Proposition 3 in [12]. However, extending this result to general bipartite graphs is non-trivial and requires dealing with possibly multiple solutions to the fixed point equation. We give a full proof in Section 4.3.

4.2 Asymptotics for large graphs

We now explain how the previous results allow to deal with random graphs model as described in Section 2.2 in the large n limit. We follow the approach of [12] based on the objective method developed by Aldous and Steele [3] and adapt it to our framework.

We are now given a sequence of bipartite graphs $G_n = (C_n \cup S_n, E_n)$ with $|C_n| = n$ and $|S_n| = \lfloor \beta n \rfloor$ and a sequence of vectors $B_n = (b_c^n)_{c\in C_n}$. We are interested in computing the limit

$$\lim_{n\to\infty} \frac{M(G_n, B_n)}{|S_n|}.$$

Of course, this limit might exist only if the sequence of graphs G_n does converge. In our case, sequences of random graphs as described in Section 2.2 have been extensively studied in the random graphs literature [6] and are known to be locally tree-like: with high probability, there is no cycle in a ball of fixed radius around a vertex chosen at random. The notion of local weak convergence [3] makes rigorous the fact that the sequence of graphs we consider in this paper converges to trees. More precisely, pick a vertex in G_n uniformly at random and call it the root. Then the local neighborhood of the root converges for the local weak convergence towards a probability distribution concentrated on rooted (possibly infinite) trees.

We now need to adapt the analysis done for finite graphs in Section 4.1 to the framework of possibly infinite graphs. We start with the following simple observation: let A^* be a maximum allocation of a finite graph G with constraints B, then

$$\frac{M(G,B)}{|S|} = \frac{1}{|S|}\sum_{s\in S} \mathbf{1}(s \text{ belongs to an edge of } A^*).$$

Now the right-hand part has a very natural probabilistic interpretaion. First consider the deterministic function mapping S to $\{0,1\}$, defined by $\mathbf{1}(s$ belongs to an edge of $A^*)$. Denote by R a node taken uniformy at random in S. We can rewrite the expression as

$$\frac{M(G,B)}{|S|} = \mathbb{E}^S\left[\mathbf{1}(R \text{ belongs to an edge of } A^*)\right], \tag{9}$$

where $\mathbb{E}^S$ is the expectation with respect to the uniform distribution over the set S. The local weak convergence directly tells us that this expectation converges to the corresponding expectation on the infinite graph provided the function $\mathbf{1}(R$ belongs to an edge of $A^*)$, which takes as input a bipartite graph $G = (C \cup S, E)$ rooted at $R \in S$ and a vector B, is continuous (with respect to the adequate topology for local weak convergence). This fact is not at all obvious and follows from [17] and [12]. Note first that the global operator $\mathcal{P}_G$ is perfectly well-defined for any locally finite graph, i.e. a graph in which each node has a finite degree. However, in an infinite setting the fixed point equation $I = \mathcal{P}_G(I)$ might not have a solution, so that it is not possible to extend the expression (8) to the infinite framework. However, Proposition 5 in [12] shows that our Theorem 1 extends nicely to the infinite framework as follows: dividing by $|S|$ the expression of F^S, we get

$$\begin{aligned}
\mathcal{F}^S(I) &= \frac{1}{|S|}F^S(I)\\
\mathcal{F}^S(I) &= \mathbb{E}^S\left[\mathbf{1}(|I_{\overrightarrow{\partial R}}| > 0)\right]\\
&\quad + \frac{|C|}{|S|}\mathbb{E}^C\left[b_R \mathbf{1}\left(\sum_{s\in\partial R} \mathcal{P}_{s\to R}(I) > b_R\right)\right]
\end{aligned}$$

where $\mathbb{E}^S$ has been defined in (9) and $\mathbb{E}^C$ is the expectation with respect to R being taken uniformly at random in the set C. For the sequence of graph G_n that we are interested in, we have $\frac{|Cn|}{|S_n|} \to \frac{1}{\beta}$ and $\mathbb{E}^{S_n}$ and $\mathbb{E}^{C_n}$ converges locally weakly to $\mathbb{E}^S$ and $\mathbb{E}^C$ on the (possibly infinite) limit graph G. We are now ready to extend Theorem 1 to the following proposition:

PROPOSITION 3. *For a sequence of bipartite graphs G_n converging locally weakly to a tree G with root R, we define*

$$\begin{aligned}
\mathcal{F}^S(I) &= \mathbb{E}^S\left[\mathbf{1}(|I_{\overrightarrow{\partial R}}| > 0)\right] \\
&\quad + \frac{1}{\beta}\mathbb{E}^C\left[b_R \mathbf{1}\left(\sum_{s\in\partial R} \mathcal{P}_{s\to R}(I) > b_R\right)\right]
\end{aligned} \tag{10}$$

Then we have

$$\lim_{n\to\infty} \frac{|M(G_n,B_n)|}{|S_n|} = \inf_{I=\mathcal{P}_G\circ\mathcal{P}_G(I)} \mathcal{F}^S(I), \tag{11}$$

where the fixed point I should not depend on the root R of G.

We can now use the markovian nature of the limiting tree in our case to compute the right expression in (11). Indeed, using the recursive structure of the tree, the fixed point equation $I = \mathcal{P}_G \circ \mathcal{P}_G(I)$ becomes a simple fixed point equation on probability distributions. More precisely, since G is now random, the variables $I_{x\to y}$ are also random and, being $\{0,1\}$-valued, are Bernoulli random variables. Moreover all the messages incoming to a vertex are independent thanks to the branching property of the limiting tree G. For simplicity, we consider the single class model, so that the limiting tree is a simple branching process with two types of nodes: nodes in S have a fixed degree d in the tree, while nodes in C have a random degree following a Poisson distribution with mean $d\beta$. In this case, we have two types of messages -those going from a vertex in S to a vertex in C and those going from a vertex in C to a vertex in S- and we need to compute their probability distributions, i.e. the corresponding parameter of the Bernoulli random variable associated to each of these messages. We denote by $I^{C\to S}$

and $I^{S\to C}$ Bernoulli random variables with respective parameter $p_{C\to S}$ and $p_{S\to C}$ corresponding to messages from C to S and S to C respectively. The values of $p_{C\to S}$ and $p_{S\to C}$ are now given by the fixed point equation $I = \mathcal{P}_G \circ \mathcal{P}_G(I)$ which gives here the following fixed point equation for the distributions of $I^{C\to S}$ and $I^{S\to C}$:

$$\begin{aligned} I^{S\to C} &\stackrel{d}{=} \mathbf{1}\left(\sum_{i=1}^{d-1} I_i^{C\to S} < 1\right) \\ I^{C\to S} &\stackrel{d}{=} \mathbf{1}\left(\sum_{i=1}^{\tilde{D}} I_i^{S\to C} < \tilde{B}\right), \end{aligned}$$

where the $I_i^{C\to S}$'s (resp. $I_i^{S\to C}$'s) are i.i.d. with the same law as $I_i^{C\to S}$ (resp. $I_i^{S\to C}$), $\tilde{D}$ is a Poisson random variable with parameter $d\beta$ (so that $\tilde{D}+1$ corresponds to the edge-biased degree distribution of a random content) and $\tilde{B}$ is a Poisson random variable with parameter λ (corresponding to the number of requests for a content).

Taking the expectation of these equations give:

$$\begin{aligned} p_{S\to C} &= (1-p_{C\to S})^{d-1} \\ p_{C\to S} &= \mathbb{P}\left(\mathrm{Bin}(\tilde{D}, p_{S\to C}) < \tilde{B}\right), \end{aligned}$$

where $\mathrm{Bin}(n,p)$ is a Binomial random variable with parameters n and $p \in [0,1]$. We can now use (10) to compute the function $\mathcal{F}^S$ as follows:

$$\begin{aligned} \mathcal{F}^S(I) &= \mathbb{E}\left[\mathbf{1}(\mathrm{Bin}(d, p_{C\to S}) > 0)\right] \\ &\quad + \frac{1}{\beta}\mathbb{E}\left[B\mathbf{1}\left(\mathrm{Bin}(\tilde{D}, p_{S\to C}) > \tilde{B}\right)\right], \end{aligned}$$

where we used the same notation as above and used the fact that the degree of a node in S (resp. C) is d (resp. a random Poisson variable with mean $d\beta$). It is convenient at this stage to introduce the following function from $[0,1]^2 \to \mathbb{R}$:

$$\begin{aligned} \mathcal{F}^S(p,q) &= \mathbb{E}\left[\mathbf{1}(\mathrm{Bin}(d, p) > 0)\right] \\ &\quad + \frac{1}{\beta}\mathbb{E}\left[B\mathbf{1}\left(\mathrm{Bin}(\tilde{D}, q) > \tilde{B}\right)\right], \end{aligned}$$

so that $\mathcal{F}^S(I) = \mathcal{F}^S(p_{C\to S}, p_{S\to C})$. We can now specialize Proposition 3 to our setting and obtain:

THEOREM 2. *For a sequence of random graphs G_n as described in Section 2.2, we have*

$$\lim_{n\to\infty} \frac{M(G_n, B_n)}{|S_n|} = \inf \mathcal{F}^S(p,q),$$

where the infimum is taken over pairs (p,q) satisfying

$$\begin{aligned} p &= \mathbb{P}\left(Bin(\tilde{D}, q) < \tilde{B}\right) \\ q &= (1-p)^{d-1}, \end{aligned}$$

where $(\tilde{D}+1, \tilde{B})$ is distributed as the edge-biased joint number of (replicas, request) of a random content, i.e. $(\tilde{D}+1, \tilde{B})$ is distributed as the joint number of (replicas, request) of the content adjacent to an edge chosen uniformly *at random among all edges of the graph.*

4.3 Proof of Theorem 1

The proof of Theorem 1 requires a basic result of graph theory. Note that in Equation (7), the left-hand side is the maximum size of an allocation whereas the right-hand side is a minimum. This kind of min-max relations is ubiquitous in matching theory [13]. In particular, for finite bipartite graphs, the size of a maximum allocation equals the minimum weight of a vertex cover. Recall that a vertex cover of a graph $G = (V, E)$ is a set of vertices $U \subset V$ such that any edge of G is incident to at least one vertex in U. The weight of a vertex cover is simply the sum of the weights of the vertices (in our case, the weights are the b_v's, equal to 1 if $v \in S$ and to b_c if $v = c \in C$.). The Koenig-Hall's theorem [13] for bipartite graphs states

$$M(G,B) = \min_{X\subseteq C}\left(|\partial X| + \sum_{c\in\overline{X}} b_c\right) \tag{12}$$

where ∂X is the set of vertices with a neighbor in X and $\overline{X} = C \setminus X$. Clearly isolated vertices do not contribute to $M(G,B)$ and can be ignored so that $\overline{X}\cup\partial X$ is a vertex cover of G (because $\overline{\partial X} \subset \partial(\overline{X})$) with weight $|\partial X| + \sum_{c\in\overline{X}} b_c$.

The proof of Theorem 1 follows then from the two following steps: we first show how to construct from $I = \mathcal{P}_G(I)$ a vertex cover with weight $F^S(I)$ which is by previous min-max relation an upper bound for $M(G,B)$. Then, we show how to construct from a particular maximum allocation A a fixed point $I^A = \mathcal{P}_G(I^A)$ such that $F^S(I^A) = |A| = M(G,B)$, so that $M(G,B)$ is an upper bound for $\inf_{I=\mathcal{P}_G(I)} F^S(I)$.

4.3.1 To each fixed point a vertex cover

For any $I \in \{0,1\}^{\vec{E}}$, we consider the following subset V^I of vertices of G: for $s \in S$, $c \in C$,

$$\begin{aligned} s \in V^I &\Leftrightarrow |I_{\vec{\partial s}}| > 0 \\ c \in V^I &\Leftrightarrow |I_{\vec{\partial c}}| > b_c. \end{aligned}$$

LEMMA 2. *If $I = \mathcal{P}_G(I)$, then the associated subset of vertices V^I defined above is a vertex cover of G.*

PROOF. Towards a contradiction, suppose there exists an edge $(cs) \in E$, with $c \in C$ and $s \in S$, which is not covered by V^I, i.e. such that $s \notin V^I$ and $c \notin V^I$. From the definition of V^I, the fact that $s \notin V^I$ implies that $|I_{\vec{\partial s}}| = 0$ so that in particular $I_{c\to s} = 0$ and then, using the fact that $I = \mathcal{P}_G(I)$, we get $I_{s\to c} = \mathbf{1}(|I_{\vec{\partial s}}| - I_{c\to s} < 1) = 1$. On the other hand, $c \notin V^I$ implies that $|I_{\vec{\partial c}}| \le b_c$ so that $I_{c\to s} = \mathbf{1}(|I_{\vec{\partial c}}| - I_{s\to c} < b_c) = 1$, a contradiction. □

It only remains to check that the weight of V^I is equal to $F^S(I)$. This is easily obtained by noting that the term $\sum_{s\in S} \mathbf{1}\left(|I_{\vec{\partial s}}| > 0\right)$ in $F^S(I)$ is the total weight of the vertices in $S\cap V^I$, and the term $\sum_{c\in C} b_c \mathbf{1}\left(|I_{\vec{\partial c}}| > b_c\right)$ in $F^S(I)$ is the total weight of the vertices in $C\cap V^I$. Since for finite bipartite graphs, the weight of any vertex cover is an upper bound on $M(G,B)$, it follows that :

$$M(G,B) \le \inf_{I=\mathcal{P}_G(I)} F^S(I) \tag{13}$$

4.3.2 To a maximum allocation a fixed point

In this section we want to find an $I \in \{0,1\}^{\vec{E}}$ that is invariant by $\mathcal{P}_G$ and such that $F^S(I) = M(G,B)$.

We start with equation (12). Note that there are no edges between vertices in X and $\overline{\partial X}$ (but there are possibly edges between $\overline{X}$ and ∂X). In particular, if A^X (resp. $A^{\overline{X}}$) is an allocation on $G[X\cup\partial X]$ (resp. $G[\overline{X}\cup\overline{\partial X}]$), the induced

graph of G on the set of vertices $X \cup \partial X$ (resp. $\overline{X} \cup \overline{\partial X}$), then $A^X \cap A^{\overline{X}} = \emptyset$ so that $A = A^X \cup A^{\overline{X}}$ is an allocation of G. Let X be a subset of C that achieves the minimum in equation (12). Let A^X (resp. $A^{\overline{X}}$) be a maximum allocation on $G[X \cup \partial X]$ (resp. $G[\overline{X} \cup \overline{\partial X}]$). Then $|A^X| \leq |\partial X|$ and $|A^{\overline{X}}| \leq \sum_{c \in \overline{X}} b_c$. There is a vertex cover of $G[X \cup \partial X]$ with weight $|A^X|$ and a vertex cover of $G[\overline{X} \cup \overline{\partial X}]$ with weight $|A^{\overline{X}}|$. The union of these vertex covers is a vertex cover of G, so that $|A^X| + |A^{\overline{X}}| \geq M(G,B)$ and finally $|A^X| = |\partial X|$ and $|A^{\overline{X}}| = \sum_{c \in \overline{X}} b_c$. Hence we defined a maximum allocation A such that with $I^A_{c \to s} = I^A_{s \to c} = \mathbf{1}((cs) \in A)$, we have

$$\forall s \in \partial X, \qquad \sum_{c \in \partial s} I^A_{c \to s} = 1$$
$$\forall c \in \overline{X}, \qquad \sum_{s \in \partial c} I^A_{s \to c} = b_c,$$

and for $u \in \overline{X}$ and $v \in \partial X$, we have

$$I^A_{u \to v} = I^A_{v \to u} = 0. \tag{14}$$

We now define the sequence of messages $I^{A,k}$ along the directed edges of G in the following manner: $I^{A,0} = I^A$ and for $k \geq 0$, $I^{A,k+1} = \mathcal{P}_G(I^{A,k})$.

LEMMA 3. *For all $u \in \partial X \cup \overline{X}$, the sequences $(I^{A,k}_{u \to v})$ are non-increasing in k for all v. For all $u \in X \cup \overline{\partial X}$, the sequences $(I^{A,k}_{u \to v})$ are non-decreasing in k for all v.*

PROOF. Note that if $u \in \partial X \cup \overline{X}$, then $|I^A_{\overrightarrow{\partial u}}| = b_u$ so that $I^{A,1}_{u \to v} = \mathbf{1}\left(|I^A_{\overrightarrow{\partial u}}| - I^A_{v \to u} < b_u\right) = I^A_{v \to u} = I^A_{u \to v}$. Now if $u \in X \cup \overline{\partial X}$, then we simply have $I^{A,1}_{u \to v} \geq I^A_{u \to v}$. In particular if $u \in \overline{X}$ and $v \in \partial X$, we have $I^{A,1}_{u \to v} = I^{A,1}_{v \to u} = I^A_{u \to v} = I^A_{v \to u} = 0$. The lemma follows then by induction on k. Indeed, the induction hypothesis implies that for $u \in \overline{X}$ and $v \in \partial X$, we have $I^{A,k}_{u \to v} = I^{A,k}_{v \to u} = 0$ by (14). Hence the $(k+1)$-th messages from $u \in \partial X \cup \overline{X}$ to v are obtained from the k-th messages from $v \in X \cup \overline{\partial X}$ by applying the decreasing map $\mathcal{P}_G$ and vice-versa. □

Lemma 3 allows to define $I^{A,\infty} = \lim_k I^{A,k}$ and to get the following inequalities:

$$\forall u \in \partial X \cup \overline{X}, \qquad I^{A,\infty}_{u \to v} \leq I^A_{u \to v},$$
$$\forall u \in X \cup \overline{\partial X}, \qquad I^{A,\infty}_{u \to v} \geq I^A_{u \to v}.$$

Moreover we clearly have $I^{A,\infty} = \mathcal{P}_G(I^{A,\infty})$. We now compute $F^S(I^{A,\infty})$ that we decompose in two terms:

$$\begin{aligned}\Sigma_1 &= \sum_{s \in S} \mathbf{1}\left(|I^{A,\infty}_{\overrightarrow{\partial s}}| > 0\right) \\ &= |\partial X| + \sum_{s \in \overline{\partial X}} \mathbf{1}\left(|I^{A,\infty}_{\overrightarrow{\partial s}}| > 0\right),\end{aligned}$$

and

$$\begin{aligned}\Sigma_2 &= \sum_{c \in C} b_c \mathbf{1}\left(|I^{A,\infty}_{\overrightarrow{\partial c}}| > b_c\right) \\ &= \sum_{c \in \overline{X}} b_c \mathbf{1}\left(|I^{A,\infty}_{\overrightarrow{\partial c}}| > b_c\right).\end{aligned}$$

Note that for any $c \in \overline{X}$, we have $|I^{A,\infty}_{\overrightarrow{\partial c}}| \geq b_c$. Indeed if $|I^{A,\infty}_{\overrightarrow{\partial c}}| > b_c$, then $I^{A,\infty}_{c \to s} = 0$ for all s, whereas if $|I^{A,\infty}_{\overrightarrow{\partial c}}| = b_c$, then $I^{A,\infty}_{c \to s} = I^{A,\infty}_{s \to c}$. Hence we have

$$\sum_{c \in \overline{X}} b_c \mathbf{1}\left(|I^{A,\infty}_{\overrightarrow{\partial c}}| = b_c\right) = \sum_{s \in \overline{\partial X}} \mathbf{1}\left(|I^{A,\infty}_{\overrightarrow{\partial s}}| > 0\right)$$

so that we get,

$$\begin{aligned}\Sigma_1 + \Sigma_2 &= |\partial X| + \sum_{c \in \overline{X}} b_c \mathbf{1}\left(|I^{A,\infty}_{\overrightarrow{\partial c}}| \geq b_c\right) \\ &= |\partial X| + \sum_{c \in \overline{X}} b_c = M(G,B).\end{aligned}$$

As a consequence, we directly get

$$M(G,B) \geq \inf_{I = \mathcal{P}_G(I)} F^S(I),$$

which with (13) concludes the proof.

5. PERFORMANCE OF REPLICATION POLICIES

We now apply the previous result to the model with K content classes, where class i contains a fraction α_i of all contents and the number of requests for each content in this class is Poisson with parameter λ_i. Recall that d is the degree of all servers and β is the ratio of number of servers to number of contents.

We consider here the couple (G,B), of the infinite graph G together with the corresponding vector of number of requests B, that is the limit in local weak sense of sequences of graphs drawn from the multi-class model presented in Section 2.2. Hence, we assume that the number of replicas for each type i content is also Poisson, with parameter $d\beta_i$ for non-negative β_i verifying $\sum_{i=1}^K \alpha_i \beta_i = \beta$. A particular example is the proportional replication strategy mentioned in Section 3, for which $\beta_i = \lambda_i / \rho$.

We express performance in terms of the asymptotic inefficiency $\gamma_{G,B} := \min(\rho, 1) - f$, where f is the limiting fraction of matched servers in a maximum allocation. Indeed, the asymptotic behavior of γ_{G_n,B_n} for (G_n, B_n) following the multi-class model will almost surely be given by $\gamma_{G,B}$, according to Theorem 2.

The explicit expression for $\gamma_{G,B}$ can be obtained straightforwadly from Theorem 2, however it is hard to gain insights from the expression obtained. Thus, we look at the behavior as d tends to infinity, as we did in Section 3 for the candidate strategies:

PROPOSITION 4. *Under the multi-class model, in which within class i the number of requests is Poisson with mean λ_i and the number of replicas is Poisson with mean $d\beta_i$, and assuming that $\beta_i > 0$ for all $i \in 1, \ldots, K$, the asymptotic inefficiency $\gamma_{G,B}$ verifies*

$$\frac{\log(\gamma_{G,B})}{d} \overset{d \to \infty}{\to} \begin{cases} -\inf_i \beta_i \ \text{if } \rho < 1, \\ \log\left(\sum_{i=1}^K \frac{\alpha_i \beta_i}{\beta} e^{-\lambda_i}\right) \ \text{if } \rho > 1, \\ \max\left(-\inf_i \beta_i, \log\left(\sum_{i=1}^K \frac{\alpha_i \beta_i}{\beta} e^{-\lambda_i}\right)\right) \\ \text{if } \rho = 1. \end{cases} \tag{15}$$

First-of-all, it is quite remarkable in the proposition above that the asymptotic inefficiency as $d \to \infty$ is explained only by very local and individual situations. Indeed, $\exp(-d\beta_i)$ is the probability that a content of class i is not replicated

in any server cache, and similarly $\left(\sum_{i=1}^{K} \theta_i e^{-\lambda_i}\right)^d$ is the probability that a server stores only replicas of contents which have not been requested at all. Hence, asymptotically, the only noticeable cause for inefficiencies in under-loaded regime (when anyway not all servers can be busy) is contents that are stored nowhere; in over-loaded regime (when it is not possible to satisfy all requests) the only visible inefficiency comes from servers that store only unrequested contents; and in critically loaded systems, inefficiency is due almost only to the dominant of these two effects.

Let us then first comment the implications of this result before turning to its proof. First, we observe an exponential decay of the inefficiency in d for any positive β_i's, *even* at criticality. This is in sharp contrast with the performance of online matching as discussed in Section 3, for which inefficiency decays polynomially in d for $\rho = 1$. It is also a robustness property, in that this qualitative behaviour does not depend on the precise values of the β_i's.

Second, it is possible to pick replication ratios β_i that improve upon proportional replication (for which $\beta_i = \lambda_i/\rho$). Indeed, it follows at once from (15) that in under-load $\rho < 1$, for large d, inefficiency is minimized by setting $\beta_i \equiv \beta$ for all i, i.e. by not discriminating replication between classes. It is also easy to show that in over-load $\rho > 1$, the corresponding exponent in (15) is minimized by shifting replicas towards the most popular class i^* such that $\lambda_{i^*} = \max_i \lambda_i$. As for the critical case, it can be shown that the corresponding exponent is minimized by equalizing all the coefficients β_i for $i \neq i^*$ in such a way that the two expressions $-\inf_i \beta_i$ and $\log(\sum_i (\alpha_i \beta_i)/\beta e^{-\lambda_i})$ are equal. We summarize these statements by the following

COROLLARY 1. *Under the multi-class model, we have the following optimal asymptotic inefficiencies:*

$$\lim_{d\to\infty} \frac{1}{d} \log(\gamma_{G,B}) = \begin{cases} -\beta & \text{if } \rho < 1, \\ -\sup_i \lambda_i & \text{if } \rho > 1, \\ -\beta^- & \text{if } \rho = 1, \end{cases} \tag{16}$$

where β^- is the unique solution x in $[0, \beta]$ to the equation

$$x = \beta \frac{e^{-x} - e^{-\lambda_{i^*}}}{\sum_i \alpha_i \left(e^{-\lambda_i} - e^{-\lambda_{i^*}}\right)}.$$

The corresponding optimal replication factors β_i are given by

$$\begin{array}{ll} \beta_i \equiv \beta & \text{if } \rho < 1, \\ \beta_{i^*} = \frac{\beta}{\alpha_{i^*}},\ \beta_i = 0,\ i \neq i^* & \text{if } \rho > 1, \\ \beta_{i^*} = \frac{\beta - \beta^-(1-\alpha_{i^*})}{\alpha_{i^*}},\ \beta_i = \beta^-,\ i \neq i^* & \text{if } \rho = 1. \end{array} \tag{17}$$

Figure 1 illustrates the inefficiency exponents for various replication policies in a two-class scenario, with two classes of equal sizes ($\alpha_1 = \alpha_2$), respective popularities $\lambda_1 = 3$ and $\lambda_2 = 1$, and the content/server ratio therefore governs the load of the system according to $\rho = 2/\beta$. the fraction of replicas that correspond to a content of class i is given by $\theta_i = \alpha_i \beta_i / \beta$. As we vary θ_1, we span replication policies from uniform at $\theta_1 = 1/2$ to proportional at $\theta_1 = 0.75$ to extreme unbalance at $\theta_1 = 1$. The curves represent the inefficiency exponents for various values of d, namely $d = 2, 5, 15$. We also represent the optimal value of θ_1 and the corresponding exponent in the limit $d \to \infty$, as characterized in the previous corollary.

We observe in particular how the optimal value for finite d approaches this limiting value: for d as small as 15, the asymptotic evaluations are already reasonably accurate. Note however that for the super-critical case, one should not take $\theta_2 = 0$ for any finite d, as is illustrated by the drop in the exponent of the right-most curve.

PROOF. (of Proposition 4): The fraction of matched servers is given by Theorem 2 and we need to explicitely compute the function $\mathcal{F}^S(p,q)$ in our setting. We still denote by $(\tilde{D}+1, \tilde{B})$ the edge-biased joint number of (replicas,requests) of a random content. Under the present assumptions, with probability α_i, the pair $(\tilde{D}+1, \tilde{B})$ is distributed as two independent Poisson random variables with respective means $(\beta_i d, \lambda_i)$. Hence a simple computation shows that

$$\begin{array}{rl} \mathcal{F}^S(p,q) = & 1 - (1-q)^d \\ & + \sum_i \frac{\alpha_i \lambda_i}{\beta} \mathbb{P}(\mathrm{Poi}(\beta_i d p) \geq \mathrm{Poi}(\lambda_i) + 2), \end{array} \tag{18}$$

where $\mathrm{Poi}(\lambda)$ is a Poisson random variable with mean λ and all Poisson random variables appearing in the expression are independent. The fraction of matched servers is given by taking the infimum of this expression over p and q satisfying:

$$q = (1-p)^{d-1}, \tag{19}$$

and

$$p = \frac{1}{\beta} \sum_i \alpha_i \beta_i \mathbb{P}(\mathrm{Poi}(\beta_i d q) < \mathrm{Poi}(\lambda_i)). \tag{20}$$

We now consider the fixed points of (19,20) in the regime of large d. First, if p is bounded away from zero, by (19), q is exponentially small in d. Plugging this into (20), we find that there is indeed a fixed point such that

$$\begin{array}{rl} p & \sim \sum_i \frac{\alpha_i \beta_i}{\beta} \left(1 - e^{-\lambda_i}\right), \\ q & \sim \left(\sum_i \frac{\alpha_i \beta_i}{\beta} e^{-\lambda_i}\right)^{d(1+\mathrm{o}(1))}. \end{array} \tag{21}$$

Next consider the case where p goes to zero with d. Then necessarily by (20), qd is large, and $p = e^{-\Theta(qd)}$. Plugging this into (19) we obtain

$$q = (1 - e^{-\Theta(qd)})^{d-1}.$$

Assume then that $qd = \mathrm{O}(\log(d))$. This would entail that $q \to 0$, and hence $(1-p)^d \to 0$; the corresponding evaluation (18) would be equivalent to $1 + \sum_i \alpha_i \lambda_i / \beta$ which is larger than 1, and hence cannot be the minimal evaluation. We can thus assume $qd \gg \log(d)$. Then by (20), it follows that $p = \mathrm{o}(1/d)$, so that by (19), $q \sim 1$. Thus the only other meaningful fixed point to consider satisfies

$$\begin{array}{rl} p = & \sum_i \frac{\alpha_i \beta_i}{\beta} e^{-\beta_i d(1+\mathrm{o}(1))} = e^{-\inf_i \beta_i d(1+\mathrm{o}(1))}, \\ q = & 1 - e^{-\inf_i \beta_i d(1+\mathrm{o}(1))}. \end{array} \tag{22}$$

It remains to evaluate Expression (18) at the two meaningful fixed points (21,22). Considering first (21), the last term in (18) is of order $(dq)^2$, which is negligible compared to the first term $(1-p)^d$. This yields the first evaluation

$$f_1 = 1 - \exp\left(d \log(\sum_i \frac{\alpha_i \beta_i}{\beta} e^{-\lambda_i})\right). \tag{23}$$

Considering next plugging (22) into (18). The term $(1-p)^d$ and the last term of (18) are equivalent to 1 and ρ respectively, up to corrections of order $e^{-d \inf_i \beta_i (1+\mathrm{o}(1))}$. This concludes the proof. □

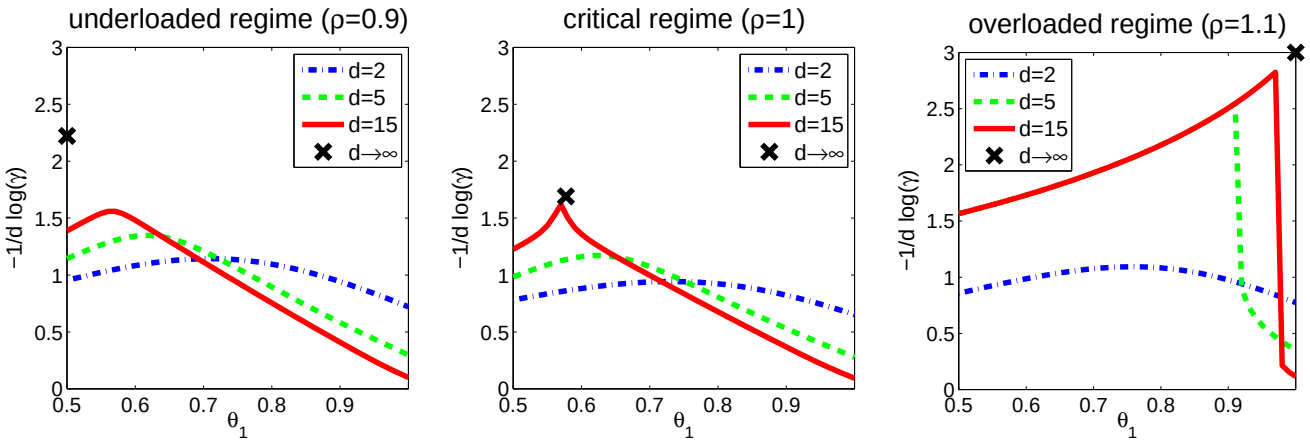

Figure 1: Inefficiency exponential-decay exponents as a function of the fraction of storage space allocated to contents of the first class in under-loaded, critical, and over-loaded regimes.

6. RELATED WORK

The question of how to replicate content in distributed systems is related to the general problem of *facility location* (see e.g. Vazirani [22] Chapter 24). The latter has received considerable attention from the standpoint of algorithmic complexity and approximability. The version that we consider here is atypical in that it features capacity constraints on the locations (the servers), and stochastic demand. Also, we aim at characterizing simple, easily implementable strategies with good performance for practical workloads rather than placement algorithms with low worst-case complexity.

There is a rich literature on cache management strategies motivated by server memory and web cache management, one abstract version of which is the so-called paging problem (see e.g. Albers [2]). In this context, the main focus has been on characterization of hit rates, focusing on the temporal properties of streams of requests. Capacity limits at the servers are typically not considered in these models, while they are essential for the application scenarios we consider.

Our present motivation, namely efficient use of servers' bandwidth through adequate replication, has been considered in the specific context of Peer-to-Peer systems, in a number of recent papers [21, 8, 18, 19, 23, 24]. In [21, 8], an argument is made for the proportional replication policy based on the analysis of delay in a queueing model of performance. More recently, [19] also argue in favour of proportional replication by considering a loss network model of performance. [18] and [8] propose replication policies that are oblivious to content popularity. The first considers stochastic delay performance models, and the second deterministic conditions on request arrivals to guarantee feasibility of service. The two articles [23, 24] are closer to our motivation in that they revisit the proportional placement strategy, to which they propose alternatives. Their modeling approach however significantly differs from ours.

The main result on asymptotic characterization of matching density, Theorem 2, is a direct consequence of the recent paper [12], itself building on recent works [7, 17] on the rigorous use of the so-called *cavity method* of statistical physics for the characterization of asymptotics in large random graphs. The version we provide for finite graphs, Theorem 1, is novel and sheds new light on the particular form of the formula in Theorem 2. It also hints at possible generalizations of the cavity method to bipartite graphs, whereas it has been rigorously applied in the context of matchings only to tree-like graphs so far.

Finally, there is a rich body of literature on load balancing algorithms, which in our context would correspond to the question of forming a particular matching, given content replication. One strand of research has considered simple one shot comparisons of several assignments, e.g. the power of two choices paradigm as surveyed in [14]. A more recent strand [9], [10], has considered the so-called Cuckoo-hashing method, by which requests get assigned irrespective of whether the target server is free or not; if not, the current job at the server is pushed to another server, and so on. These two approaches are in a sense intermediate between the online method we considered in Section 3 and an optimal assignment as characterized and discussed in Sections 4-5. Their performance has however not been studied in contexts with heterogeneity both in the replications and numbers of requests. We view such a study as a promising future work, which could build on our present analysis of optimal matching density .

7. CONCLUSION

Motivated by the performance of large-scale distributed server systems, we developed an analysis of optimal matching performance for heterogeneous loads and replication policies. Our results show (i) robustness of optimal matching performance to replication strategies, with an inefficiency that is exponentially small in the server storage capacity; (ii) possibility to improve upon proportional replication, with explicit identification of target replication strategies, depending on the criticality of the system load; and (iii) the need to go beyond simple greedy matching in order to approach these ideal matching inefficiencies. On the methodological side, we build on recent advances on the so-called cavity method, and use it to obtain novel performance formulas.

Moreover, we extend previous characterizations of maximum matchings from finite trees to finite bipartite graphs. This points the way to further extensions of the cavity method.

8. ACKNOWLEDGMENTS

The second author acknowledges the support of the French Agence Nationale de la Recherche (ANR) under reference ANR-11-JS02-005-01 (GAP project).

9. REFERENCES

[1] S. Abramsky, C. Gavoille, C. Kirchner, F. M. auf der Heide, and P. G. Spirakis, editors. *ICALP 2010*, volume 6198 of *Lecture Notes in Computer Science*. Springer, 2010.

[2] S. Albers, L. Favrholdt, and O. Giel. On paging with locality of reference. *Journal of Computer and System Sciences*, 70:145–175, 2005.

[3] D. Aldous and J. M. Steele. The objective method: probabilistic combinatorial optimization and local weak convergence. In *Probability on discrete structures*, volume 110 of *Encyclopaedia Math. Sci.*, pages 1–72. Springer, Berlin, 2004.

[4] R. P. Anstee. A polynomial algorithm for b-matchings: an alternative approach. *Inform. Process. Lett.*, 24(3):153–157, 1987.

[5] M. Bayati, C. Borgs, J. Chayes, and R. Zecchina. Belief propagation for weighted b-matchings on arbitrary graphs and its relation to linear programs with integer solutions. *SIAM Journal on Discrete Mathematics*, 25(2):989–1011, 2011.

[6] B. Bollobás. *Random Graphs*. Cambridge University Press, 2001.

[7] C. Bordenave, M. Lelarge, and J. Salez. Matchings on infinite graphs. *ArXiv e-prints*, Feb. 2011.

[8] Y. Boufkhad, F. Mathieu, F. de Montgolfier, D. Perino, and L. Viennot. Achievable catalog size in peer-to-peer video-on-demand systems. In *Proc. of IPTPS*, 2008.

[9] M. Dietzfelbinger, A. Goerdt, M. Mitzenmacher, A. Montanari, R. Pagh, and M. Rink. Tight thresholds for cuckoo hashing via xorsat. In Abramsky et al. [1], pages 213–225.

[10] N. Fountoulakis and K. Panagiotou. Orientability of random hypergraphs and the power of multiple choices. In Abramsky et al. [1], pages 348–359.

[11] T. Kurtz. *Approximation of Population Processes*. CBMS-NSF Regional Conference Series in Applied Mathematics, 1981.

[12] M. Lelarge. A new approach to the orientation of random hypergraphs. *available at: http://www.di.ens.fr/ lelarge/*, 2012.

[13] L. Lovász and M. D. Plummer. *Matching theory*, volume 121 of *North-Holland Mathematics Studies*. North-Holland Publishing Co., Amsterdam, 1986. Annals of Discrete Mathematics, 29.

[14] M. Mitzenmacher, A. Richa, and R. Siaraman. The power two random choices: a survey of techniques and results. In *Handbook of randomized computing*, volume 1, 2001.

[15] J. Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the American Association of Artificial Intelligence National Conference on AI*, pages 133–136, Pittsburgh, PA, 1982.

[16] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[17] J. Salez. The cavity method for counting spanning subgraphs subject to local constraints. *ArXiv e-prints*, Mar. 2011.

[18] K. Suh, C. Diot, J. Kurose, L. Massoulie, C. Neumann, D. Towsley, and M. Varvello. Push-to-peer video-on-demand system: Design and evaluation. *IEEE Journal on Selected Areas in Communications*, 25(9):1706–1716, 2007.

[19] B. Tan and L. Massoulié. Optimal content placement for peer-to-peer video-on-demand systems. In *Proceedings of IEEE Infocom*, 2011.

[20] É. Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research*, 34(2):pp. 250–256, 1986.

[21] S. Tewari and L. Kleinrock. On fairness, optimal download performance and proportional replication in peer-to-peer networks. In *Proc. of IFIP Networking*, 2005.

[22] V. Vazirani. *Approximation Algorithms*. Springer, 2003.

[23] W. Wu and J. Lui. Exploring the optimal replication strategy in p2p-vod systems: Characterization and evaluation. In *Proceedings of IEEE Infocom*, 2011.

[24] Y. Zhou, T. Z. J. Fu, and D. Ming Chiu. Statistical modeling and analysis of p2p replication to support vod service. In *Proceedings of IEEE Infocom*, 2011.

10. APPENDIX:

Online algorithm analysis

Letting $u = d\frac{\beta - x}{1-y}$, we obtain

$$\begin{aligned} \dot{u} &= ue^{-u} + de^{-u} - d \\ \gamma &= \rho \wedge 1 - 1 + \frac{u(\lambda)}{d\beta}(1 - y(\lambda)) \end{aligned} \tag{24}$$

To find a good surrogate for u, note first that u is a decreasing function, because $e^u \dot{u} = u + d - de^u \leq -(d-1)u$. Also, the term ue^{-u} is always small compared to the others when d is large.

It suggests that the behavior of u for large d should be captured by the solution to the following equation:

$$\begin{aligned} \dot{v} &= d(e^{-v} - 1) \\ v(0) &= u(0) = d\beta \end{aligned} \tag{25}$$

A change of variables leads to

$$v(t) = \log\left(1 + e^{d(\beta - t)}(1 - e^{-d\beta})\right)$$

Note that, at all time, we have $v \leq u$, because the functions are continuous and $\dot{v} \leq \dot{u}$ whenever $v = u$.

We then have

LEMMA 4. *In the regime $\rho < 1$, the solution of equations (4)-(5) satisfies*

$$\gamma \leq \rho(\beta + 1)e^{-d\beta(1-\rho)}$$

PROOF. At all time, $d\beta \geq u \geq d\beta(1-\rho)$, thus $\dot{u} \leq d\left((\beta+1)e^{-d\beta(1-\rho)} - 1\right)$. We immediately obtain

$$\begin{aligned} u(\lambda) &\leq d\beta(1-\rho) + d\lambda(\beta+1)e^{-d\beta(1-\rho)} \\ \gamma &= \rho - 1 + \frac{1}{d\beta}u(\lambda)(1 - y(\lambda)) \leq \rho(\beta+1)e^{-d\beta(1-\rho)} \end{aligned}$$

□

LEMMA 5. *In the regime $\rho = 1$, the solution of equations (4)-(5) satisfies*

$$\gamma \leq \frac{\log 2}{d\beta} + \mathrm{o}\left(1/d\right)$$

PROOF. Let $c > 1$. $u(0) = d\beta$ so there exists $0 < T_d \leq \lambda$ such that $u \geq c \log d$ on $[0, T_d]$ and $u \leq c \log d$ on $(T_d, \lambda]$ for large enough d. Furthermore, $\dot{u} \geq -d$, so we always have $T_d \geq \beta - \frac{c \log d}{d}$ for d large enough. Then, on $[0, T_d]$, we have

$$\dot{u} \leq \beta d^{1-c} + d(e^{-u} - 1) \leq \beta d^{1-c} + \dot{v}$$

and, on $(T_d, \rho]$,

$$\dot{u} \leq c \log d + \dot{v}$$

We obtain an upper-bound on $u(\lambda)$:

$$\begin{aligned} u(\lambda) &\leq v(\lambda) + T_d \beta d^{1-c} + (\lambda - T_d)c \log d \\ &\leq \log\left(2 - e^{-d\beta}\right) + \beta d^{1-c}\lambda + \frac{c^2 \log^2 d}{d} \end{aligned}$$

As $c > 1$,

$$\begin{aligned} \gamma &\leq \frac{\log 2}{d\beta} + d^{-c}\rho + \frac{c^2 \log^2 d}{d^2\beta^2} \\ &= \frac{\log 2}{d\beta} + \mathrm{o}(1/d) \end{aligned}$$

□

LEMMA 6. *In the regime $\rho = 1$, the solution of equations (4)-(5) satisfies*

$$\gamma \geq \frac{\log 2}{d\beta} + \mathrm{o}\left(1/d\right)$$

PROOF. As $y + x \leq \lambda$, it follows that $y \leq \beta\gamma \leq \frac{\log 2}{d} + \mathrm{o}(1/d)$. Hence,

$$\begin{aligned} \gamma &\geq \frac{1}{d\beta}\log\left(2 - e^{-d\beta}\right)\left(1 - \frac{\log 2}{d} + \mathrm{o}(1/d)\right) \\ &= \frac{\log 2}{d\beta} + \mathrm{o}(1/d) \end{aligned}$$

□

As upper- and lower-bound coincide, we actually have that $\gamma = \frac{\log 2}{d\beta} + \mathrm{o}(\frac{1}{d}) \sim_{d\infty} v(\lambda)$.

LEMMA 7. *In the regime $\rho > 1$, the solution of equations (4)-(5) satisfies*

$$\gamma \leq \frac{\log 2}{d^2\beta}e^{-(d-1)\beta(\rho-1)}(1 + \mathrm{o}(1))$$

PROOF. We already know that $u(\beta) = \frac{\log 2}{d\beta} + \mathrm{o}(\frac{1}{d})$, so we can focus on the time interval $[\beta, \lambda]$. For d large enough and for all $t \geq \beta$, as u is decreasing, u is strictly less than 1. Then,

$$\dot{u} \leq u + d(1 - u + \frac{u^2}{2} - 1) = -(d-1)u + d\frac{u^2}{2}$$

As u is actually much smaller than 1 in the range $[\beta, \lambda]$, we can guess that the influence of the term in u^2 will be small compared to that of the term in u, and that we did not lose much by neglecting higher order terms.

Let z such that $z(\beta) = u(\beta)$ and $\dot{z} = -(d-1)z + d\frac{z^2}{2}$. At $u = z$, we have $\dot{u} \leq \dot{z}$, so $u \leq z$ on $[\beta, \lambda]$, and z is decreasing on $[\beta, \lambda]$ for d large enough.

As long as $z \neq 0$ and $z \neq 2(1 - \frac{1}{d})$,

$$\dot{z} = -(d-1)z + d\frac{z^2}{2} \Leftrightarrow \frac{\dot{z}}{d-1}\left(\frac{1}{z - 2(1-\frac{1}{d})} - \frac{1}{z}\right) = 1$$

Integrating from λ to β, we obtain

$$\begin{aligned} \frac{1}{d-1}\left[\log\left(z - 2(1 - \frac{1}{d})\right)\right]_\beta^\lambda - \frac{1}{d-1}\left[\log z\right]_\beta^\lambda &= \lambda - \beta \\ \frac{1}{d-1}\log\frac{1 - \frac{z(\lambda)}{2(1-\frac{1}{d})}}{1 - \frac{u(\beta)}{2(1-\frac{1}{d})}} + \frac{1}{d-1}\log\frac{u(\beta)}{z(\lambda)} &= \lambda - \beta \end{aligned}$$

As $0 < z(\lambda) < u(\beta) = \frac{\log 2}{d} + \mathrm{o}(\frac{1}{d})$, the first term is $\mathrm{o}(1)$. Thus, we obtain

$$\begin{aligned} u(\lambda) &\leq z(\lambda) = u(\beta)e^{-(d-1)\beta(\rho-1) + \mathrm{o}(1)} \\ &= \frac{\log 2}{d}e^{-(d-1)\beta(\rho-1)}(1 + \mathrm{o}(1)) \end{aligned}$$

and also

$$\gamma \leq \frac{\log 2}{d^2\beta}e^{-(d-1)\beta(\rho-1)}(1 + \mathrm{o}(1))$$

□