Dynamic Content Allocation for Cloud-assisted Service of Periodic Workloads

György Dán *, Niklas Carlsson†
* School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, gyuri@ee.kth.se
†Linköping University, Linköping, Sweden, nikca@ida.liu.se

***Abstract*—Motivated by improved models for content workload prediction, in this paper we consider the problem of dynamic content allocation for a hybrid content delivery system that combines cloud-based storage with low cost dedicated servers that have limited storage and unmetered upload bandwidth. We formulate the problem of allocating contents to the dedicated storage as a finite horizon dynamic decision problem, and show that a discrete time decision problem is a good approximation for piecewise stationary workloads. We provide an exact solution to the discrete time decision problem in the form of a mixed integer linear programming problem, propose computationally feasible approximations, and give bounds on their approximation ratios. Finally, we evaluate the algorithms using synthetic and measured traces from a commercial music on-demand service and give insight into their performance as a function of the workload characteristics.**

I. Introduction

The past decade witnessed the migration of content delivery systems from dedicated servers to shared infrastructures, to content distribution networks (CDNs) and to cloud-based content delivery platforms. CDN and cloud-based content delivery offers a number of advantages to content providers. It facilitates the fast expansion of the content catalogue without infrastructure investments. On-demand computational power can be used for scaling the content indexing, user management, and accounting workloads. Finally, bandwidth is available on-demand and can be used to serve fast varying content workloads with reduced need for an understanding of the characteristics of the system's workload.

Flexibility comes, however, at extra cost. For example, a back-of-the-envelope calculation reveals that the cost of 100Mbps dedicated upload bandwidth (sufficient to upload over 30TB of data in a month) is only a fraction of the equivalent CDN or cloud bandwidth cost. With current prices the difference exceeds an order of magnitude; e.g., \$100USD per month vs. \$3,000USD, at a CDN or cloud bandwidth price of \$0.1 USD/GB. Furthermore, as the online content market matures, content providers strive for higher quality of experience (QoE) for retaining their customers, which has proven difficult with CDN and cloud-based delivery [1].

Motivated by the traffic generated by over-the-top multimedia services and by the growing emphasis on QoE, content providers like Netflix and network operators have started to deploy dedicated servers closer to the customers. By relying on dedicated servers, owned or rented, for serving part of the content catalogue, the emerging hybrid content delivery platforms combine cloud and CDN based storage with dedicated servers and upload bandwidth. This hybrid solution provides the convenience of scalable storage for maintaining a possibly large content catalogue in the cloud or CDN, but it can also leverage the low cost and customer proximity of dedicated unmetered network bandwidth.

The traditional approach to managing the storage and bandwidth resources of dedicated in-network servers is to cache popular content. There is a wide range of cache eviction policies, from simple least recently (LRU) or least frequently used (LFU) to more sophisticated policies [2]. Improved workload models do, however, allow operators to predict content popularity [3], [4], [5], [6]. Popularity predictions in turn enable the use of prefetching instead of caching, which allows the operators to schedule the move of data to the in-network storage. Prefetching the most popular contents could also reduce the amount of data downloaded to the cache compared to LRU and LFU [7], but it cannot leverage the fluctuations of the popularity of different contents over time.

Given the availability of workload predictions, prefetching could be dynamically adapted to changing predicted popularities. Nevertheless, it is unclear whether and when dynamic content allocation could provide benefits compared to a static, average demand based prefetching scheme or compared to caching. Compared to static prefetching and to caching, the challenge lies in balancing (i) the cost of moving contents into dedicated storage, from which it can be served at a low cost, (ii) the high costs associated with the demands that cannot be served from the dedicated storage, and (iii) the opportunity loss associated with not fully using the dedicated bandwidth.

In this paper we make three important contributions to address this challenge. First, we formulate the problem of dynamically allocating contents to the dedicated storage as a finite horizon dynamic decision process, and we show that a discrete time decision process is a good approximation for piecewise stationary workloads. Second, we provide an exact solution to the discrete time decision problem in the form of a mixed integer linear programming problem. We provide computationally feasible approximations to the exact solution and provide results on their approximation ratios. Third, we validate the model and the algorithms using measured traces from a commercial on-demand music streaming system, and show how the efficiency of content allocation depends on the level of understanding of the content workload and on the amount of information available about its statistical behavior. To the best of our knowledge this is the first work that presents

an analysis and optimization of the delivery costs of a content provider using a hybrid system that combines dedicated and on-demand bandwidth to serve periodic workloads.

The rest of the paper is organized as follows. Section II describes the system model and optimization problem. Section III introduces the discrete time approximation for piecewise stationary demands. Section IV presents an exact solution and computationally feasible approximations with provable approximation ratios. Section V evaluates the performance of the proposed approximation policies. Section VI reviews related work, and Section VII concludes the paper.

II. System Model

We consider a content provider that serves a large population of users from a catalogue $\mathcal{F}$ of $F(\equiv|\mathcal{F}|)$ files. We denote the size of file $f \in \mathcal{F}$ by L_f, and consider that the provider aims to achieve an average file delivery time τ_f. The user requests for file f generate bandwidth demand $B_f(t)$ at time t. We model $B_f(t)$ as a continuous time piecewise stationary stochastic process with finite mean and variance; that is, the mean and variance are a function of time. The piecewise stationary assumption is motivated by the observed diurnal fluctuations of content workloads [8], [9]. We consider that for every stationary interval $[t_i, t_{i+1}]$ the content provider has a prediction of the average bandwidth demand $\overline{B}^i_f$ of every file f. This is a reasonable assumption, as fairly accurate predictions can be obtained based on past content popularity, as we will see later. For a set of files $\mathcal{X}$ we use the shorthand notation $\overline{B}^i_{\mathcal{X}} = \sum_{f\in\mathcal{X}} \overline{B}^i_f$.

A. Traffic Cost Model

User requests can be served from the cloud using cloud bandwidth, which is charged by volume. Alternatively, if a requested file is available in the *dedicated storage*, it can be served using *unmetered dedicated upload bandwidth*. We denote by S the amount of dedicated storage and by U the amount of unmetered dedicated bandwidth.

Given the amount of storage and unmetered bandwidth, the content provider has to choose the set of files to be stored in the dedicated storage. We denote by $\mathcal{X}(t)$ the set of files stored in the dedicated storage at time t, which in order to be feasible has to satisfy the storage constraint

$$\sum_{f\in\mathcal{X}(t)} L_f \le S, \ \forall t. \tag{1}$$

Given the set of feasible storage allocations, a storage allocation policy π defines $\mathcal{X}^\pi(t)$ as a function of the system's history up to time t and the predicted future bandwidth demands. We denote the set of all feasible allocation policies by Π.

Let us consider now the amount of cloud traffic during a time interval $T = [t_0, t_B]$. Let us denote by t^π_i, $i = 1,\ldots,I^\pi$ the i^{th} time instant when the content provider changes the allocation, $t_0 \le t^\pi_1 < \ldots < t^\pi_{I^\pi} < t_B$, and let $\mathcal{X}^\pi_i$ be the set of stored files right after t^π_i; i.e., as a result of the decision. We use the notation $\mathcal{X}^\pi_0 = \mathcal{X}^\pi(t_0)$, define $t^\pi_{I^\pi+1} = t_B$, and denote the set of files fetched upon the i^{th} decision by $A^\pi_i = \mathcal{X}^\pi_i \setminus \mathcal{X}^\pi_{i-1}$.

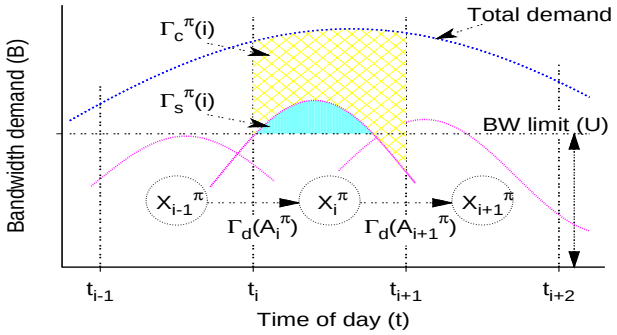

Fig. 1. The total demand for bandwidth is served from the dedicated server with bandwidth limit U and from the cloud. In interval $[t_i, t_{i+1})$, $\Gamma^\pi_s(i)$ is the data served from the cloud for files $\mathcal{X}^\pi_i$ stored on the server due to bandwidth spillover, $\Gamma^\pi_c(i)$ is the data served from the cloud for files not stored on the server. $\Gamma^\pi_d(A^\pi_i)$ is the data served from the cloud to store files on the server.

The expected cloud traffic needed to satisfy the bandwidth demand of the files not stored on the servers under policy π between two decision instants can be expressed as

$$\Gamma^\pi_c(i) = E\left[\int_{t^\pi_i}^{t^\pi_{i+1}} \sum_{f\notin\mathcal{X}^\pi_i} B_f(t)\right], \tag{2}$$

where the expectation is taken over the future demands.

The instantaneous aggregate bandwidth demand for the files stored on the dedicated servers may exceed the dedicated bandwidth, in which case the excess demand has to be served using cloud bandwidth. We refer to this traffic as *spillover* traffic. The expected spillover traffic between two decision instants can be expressed as

$$\Gamma^\pi_s(i) = E\left[\int_{t^\pi_i}^{t^\pi_{i+1}} \left(\sum_{f\in\mathcal{X}^\pi_i} B_f(t) - U\right)^+ dt\right]. \tag{3}$$

Finally, every decision of the content provider to change the set of allocated files generates cloud traffic. The cloud traffic induced by the i^{th} decision can be expressed as

$$\Gamma^\pi_d(A^\pi_i) = \sum_{f\in A^\pi_i} L_f, \tag{4}$$

and we define $\Gamma^\pi_d(A^\pi_0) = 0$. Fig. 1 illustrates the three kinds of traffic for three consecutive intervals and the bandwidth limit.

We consider that cloud traffic is charged by volume, which is the case for all major cloud providers, and denote the unit price by γ. The cloud traffic cost during an interval is then modeled as a linear function of the data served from the cloud

$$J^\pi(T, \mathcal{X}_0) = \gamma \times \sum_{i=0}^{I^\pi} \{\Gamma^\pi_d(A^\pi_i) + \Gamma^\pi_c(i) + \Gamma^\pi_s(i)\}. \tag{5}$$

In practice, the traffic cost is often a concave non-decreasing piecewise linear function of the amount of uploaded data. Our results for a linear cost function can easily be generalized to concave, piecewise linear functions.

B. Problem Formulation

While it would be natural to formulate the objective of the content provider as the minimization of $J^\pi(T, \mathcal{X}_0)$, a more insightful formulation can be obtained by converting the problem

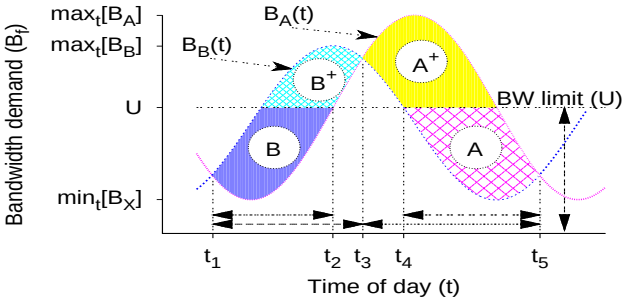

Fig. 2. For bandwidth limit U file A should be stored instead of B some time between t_2 and t_4 if area A is greater than L_A. Without bandwidth limit, the decision should be made at time t_3, if area $A \cup A^+$ is greater than L_A.

into an equivalent utility maximization problem. To see how, consider the total cost under cloud-only content delivery and subtract the actual cost for policy π. The cloud traffic savings (i.e., the traffic served from the dedicated servers between two decision instants) can be expressed as

$$\overline{\Gamma}_s^{\pi}(i) = E\left[\int_{t_i^{\pi}}^{t_{i+1}^{\pi}} \min\left(U, \sum_{f\in\mathcal{X}_i^{\pi}} B_f(t)\right) dt\right]. \quad (6)$$

The cost savings over interval T can then be expressed as

$$U^{\pi}(T,\mathcal{X}_0) = \gamma \times \sum_{i=0}^{I^{\pi}} \left\{\overline{\Gamma}_s^{\pi}(i) - \Gamma_d^{\pi}(A_i^{\pi})\right\}, \quad (7)$$

and the objective of the content provider is to find

$$\pi^* = \arg\max_{\pi} U^{\pi}(T,\mathcal{X}_0). \quad (8)$$

The policy π^* defines the decision instants $t_i^{\pi^*}$ and the set of files $\mathcal{X}_i^{\pi^*}$ to be stored at the dedicated storage upon time instant i, which defines $A_i^{\pi^*}$.

Example: To help build an intuition for the utility maximization problem, consider the example in Fig 2. There are two files, A and B, with bandwidth demands $B_A(t)$ and $B_B(t)$. The dedicated storage is enough for one file only, and file A is originally in storage. The expected bandwidth demand of file B exceeds that of file A between time t_1 and t_3. Thus, at t_1 we must decide if we should replace A with B. For a bandwidth limit of U, making this switch would at most save the cloud traffic associated with the area between the two bandwidth curves marked "B" in the figure, minus the cloud traffic L_B associated with downloading file B to storage. (If there was no bandwidth limit, the cloud traffic saving would be the sum of the areas marked "B" and "B$^+$", minus L_B.) At time t_1, the optimal policy would be to replace A with B only if $\int_{t_1}^{t_2}(\min[U,B_B(t)] - B_A(t))dt - L_B > 0$. Similarly, assuming that this switch is made, an optimal policy would select to switch back to A at some point between t_3 and t_4 if $\int_{t_4}^{t_5}(\min[U,B_A(t)] - B_B(t))dt - L_A > 0$.

The above example helps illustrate the power of the modified formulation for identifying an optimal policy. In the following we address the question what policy should the content provider use to allocate files to the dedicated storage to maximize its cost saving defined in (7).

III. Discrete-time Approximation for Piecewise Stationary Demands

The solution of the traffic cost saving maximization (8) in the framework of continuous time decision processes faces two major challenges. First, the bandwidth demands are non-stationary. Thus, even though a stationary policy would exist, it would only depend on the current system state, and would not be able to leverage predictions of the future system state. Second, as shown by equation (6), the amount of traffic served from the dedicated servers depends on the distribution of the sum of random variables, and thus an exact maximization of $\overline{\Gamma}_s^{\pi}(i)$ can be infeasible.

In the following, we show that by decomposing the bandwidth demand process into consecutive steady state intervals we can use the mean bandwidth demands for optimization at the price of minimal loss in accuracy. Motivated by this observation, in Section IV, we then show that a discrete time decision problem can provide the optimal solution.

A. Mean-value Approximation in Steady State

Let us consider a system in steady state. Users that want to download file f arrive according to a Poisson process with rate λ_f, and thus, the bandwidth demands $B_f(t)$ of the files are stationary stochastic processes. Since the average service time of file f is τ_f, the number of users that are downloading file f can be modeled by an M/G/∞ queue. The probability that there are i users downloading file f at time t is the steady state distribution of the number of users in the queue

$$p_i(f) = \frac{(\lambda_f\tau_f)^i}{i!} e^{-\lambda_f\tau_f}. \quad (9)$$

The instantaneous bandwidth demand $B_f(t)$ depends on the instantaneous number of users. Thus, even if the expected aggregate bandwidth demand $\sum_{f\in\mathcal{X}} E[B_f(t)]$ of the files stored on the server is less than the unmetered bandwidth U, in a system in steady state the spillover traffic $\Gamma_s^{\pi}(i,b)$ can be positive; effectively depending on the tail probability of the stationary bandwidth demands. Similarly, the aggregate bandwidth demand of the files stored on the dedicated servers can be less than the unmetered bandwidth U. While for a single file the bandwidth demand $B_f(t)$ can be far from its expected value $E[B_f(t)]$, in the following we show how the individual bandwidth demand distributions can be used to bound the tail and the head probability of the aggregate bandwidth demand $\sum_{f\in\mathcal{X}} B_f(t)$ for an arbitrary set $\mathcal{X}$ of files under two scenarios. The bounds of the head and tail probability in turn bound the potential (typically small) error using mean-value-based dimensioning for a time-interval during which the bandwidth demand is in steady state.

1) Server-only Data Delivery: We first consider the scenario when the users download all data from the servers managed by the content provider. In this case, the total bandwidth demand is proportional to the total number of users in the system. If the dynamics are fast compared to the rate at which decisions are made, the number of users downloading the different files can be well modeled by independent random

variables. Under this independence assumption between request rates, the total number of instantaneous users is Poisson distributed with parameter $\rho = \lambda\tau = \sum_{f\in\mathcal{X}} \lambda_f \tau_f$.

Given that the number of users is Poisson distributed with parameter $\rho = \lambda\tau$, the probability that the instantaneous aggregate bandwidth demand for the set $\mathcal{X}$ of files stored on the dedicated servers is less than the unmetered bandwidth U can be calculated as

$$P(\sum_{f\in\mathcal{X}} B_f(t) \leq U) = P(n \leq n^*) = \sum_{i=0}^{n^*} e^{\lambda\tau} \frac{(\lambda\tau)^i}{i!}, \qquad (10)$$

where we have used that the unmetered bandwidth U allows at most n^* clients to be served simultaneously by the servers.

The tail behavior of this distribution can easily be bounded using standard techniques. For example, Michel [10] have shown that the error of approximating equation (10) with the cumulative standard normal distribution $\Phi(\beta)$ is inversely proportional to the average number of simultaneous clients $\lambda\tau$ in the system; i.e.,

$$|P(n \leq n^*) - \Phi(\beta)| \leq \frac{0.8}{\sqrt{\lambda\tau}}, \qquad (11)$$

where $\beta = (n^* - \lambda\tau)/\sqrt{\lambda\tau}$. Tighter bounds, such as those recently proposed by Janssen et al. [11], do not alter the general shape of the tail behavior; only the accuracy of the estimations that the bounds provide.

An important observation here is that the probability $P(\sum_{f\in\mathcal{X}} B_f(t) \leq U)$ of under-utilizing the unmetered bandwidth at the dedicated servers decreases exponentially with the amount of unmetered bandwidth U (or equivalently, the maximum number of clients n^* that can be served simultaneously), given a fixed average fraction of the bandwidth demand $\frac{U}{\sum_{f\in\mathcal{X}} B_f(t)}$ (or fraction of clients $\frac{n^*}{\lambda\tau}$) that potentially could be served. To see this, note that $\beta = (n^* - \lambda\tau)/\sqrt{\lambda\tau} = \sqrt{\frac{n^*}{\lambda\tau}}(1 - \frac{\lambda\tau}{n^*})\sqrt{n^*}$. Hence, β scales as $\sqrt{n^*}$, for a fixed $\frac{n^*}{\lambda\tau}$ ratio. With $\Phi(\beta) \sim \int e^{\beta^2/2} d\beta$, for a fixed $\frac{n^*}{\lambda\tau}$ ratio, the probability $P(\sum_{f\in\mathcal{X}} B_f(t) \leq U)$ therefore decreases exponentially whenever $U < \sum_{f\in\mathcal{X}} E[B_f(t)]$. (Similarly, the probability $P(\sum_{f\in\mathcal{X}} B_f(t) \geq U)$ decreases exponentially whenever $U > \sum_{f\in\mathcal{X}} E[B_f(t)]$.)

To summarize, the key insight from this discussion is that already for small number of files the instantaneous steady state bandwidth demand is fairly stable, and as the number of files increases, the allocation problem gets close to deterministic. Thus, if the overall request rate (or the number of files with reasonable request rates) that can be served on the dedicated server is sufficiently large, mean-value-based dimensioning provides a good (and increasingly accurate) approximation.

2) Peer-assisted Data Delivery: As an alternative, we consider a peer-assisted system in which the content provider also leverages the users' (peers') bandwidth for distributing the content. In an ideal peer-assisted system, for each file, the bandwidth demand is $\frac{1}{\tau}$ whenever there are $i \geq 1$ users downloading simultaneously, and 0 otherwise. More efficient server-bandwidth allocation policies for peer-assisted content delivery can be used when swarms become self-sustaining [12], [13], without affecting our conclusions. Now, for every file, the probability that there is at least one active user downloading file f can be modeled as a Bernoulli distribution with probability $p_f = 1 - e^{-\lambda_f \tau_f}$, the expected bandwidth demand for a single file is $E[B_f(t)] = \frac{1}{\tau_f}(1 - e^{-\lambda_f\tau_f})$, and the total instantaneous bandwidth demand for the system is equal to the number of files with at least one active user. The bandwidth demand can thus be modeled as the sum of independent, non-identically distributed Bernoulli random variables. For the purpose of our analysis, we denote the (complementary) probability that there are no downloaders in the system by $q_f = 1 - p_f = e^{-\lambda_f\tau_f}$.

We can again provide a bound on the probability that the instantaneous demand for the set $\mathcal{X}$ of files stored on the dedicated servers is less than the unmetered bandwidth U. For this bound we rely on a result by Siegel [14], which states that the probability that the sum $Y = \sum_{f\in\mathcal{X}} y_f$ of $|\mathcal{X}|$ independent Bernoulli trials with probabilities q_f can be expressed as

$$P(Y \geq |\mathcal{X}|(\bar{q}+a)) \leq (\frac{\hat{q}}{\hat{q}+a})^{(a+\hat{q})|\mathcal{X}|} (\frac{1-\bar{q}}{1-\bar{q}-a})^{(1-\bar{q}-a)|\mathcal{X}|}, \qquad (12)$$

where $\bar{q} = \frac{E[Y]}{|\mathcal{X}|}$, $\sigma^2 = \frac{E[Y^2]-E[Y]^2}{|\mathcal{X}|}$, $\hat{q} = \frac{\sigma^2}{1-\frac{\sigma^2}{1-\bar{q}}}$, and $a > 0$.

Consider again the case when $U < \sum_{f\in\mathcal{X}} E[B_f(t)]$, and let us calculate the probability that $P(\sum_{f\in\mathcal{X}} B_f(t) \leq U)$. Note that this probability is equal to the probability that $P(Y \geq |\mathcal{X}| - n^*)$. By equating this expression to our previously defined probability $P(Y \geq |\mathcal{X}|(\bar{q}+a))$, we can express a in terms of our original variables as

$$a = 1 - \frac{n^*}{|\mathcal{X}|} - \bar{q}. \qquad (13)$$

Similarly as for the server-only case, we note that for a given $\frac{n^*}{|\mathcal{X}|}$ ratio, the probability that $P(\sum_{f\in\mathcal{X}} B_f(t) \leq U)$ decreases exponentially with $|\mathcal{X}|$ whenever $U < \sum_{f\in\mathcal{X}} E[B_f(t)]$, and hence (also in this case), mean-value-based dimensioning is a good (and increasingly accurate) approximation when the number of files is sufficiently large.

IV. Storage Allocation Policies

A. Optimality of Discrete-time Decision Problem

Given the piecewise stationary behavior of the bandwidth demands and the mean value approximation, we can now define an equivalent discrete-time decision problem for (8). The following proposition shows that if the mean value approximation is accurate then there is an optimal policy that makes updates only upon transitions between stationary regimes.

Proposition 1: Consider the continuous time decision problem (8). If the aggregate bandwidth demand $\sum_{f\in\mathcal{X}(t)} B_f(t)$ *can be approximated by* $\sum_{f\in\mathcal{X}_i} \overline{B}_f^i$ *for time* $t_i \leq t < t_{i+1}$*, then there is an optimal policy* π^* *such that* $t_i^{\pi^*} = t_i$*; that is, the set of stored files is* only *changed upon transitions between stationary regimes of the bandwidth demands.*

Proof: Consider the continuous time decision problem (8), and assume that there is an optimal policy π' for which $t_j = t_i^{\pi'} < \ldots < t_{i+k}^{\pi'} = t_{j+1}$ for some $k \geq 2$, and $\mathcal{X}_{i'}^{\pi'} \neq \mathcal{X}_{i'-1}^{\pi'}$ for $i < i' < i+k$. For a set $\mathcal{X}$ of files denote the aggregate average demand by $\overline{B}_{\mathcal{X}}^j = \sum_{f\in\mathcal{X}} \overline{B}_f^j$, and let $\overline{B}_U^j(\mathcal{X}) = \min\left(U, \overline{B}_{\mathcal{X}}^j\right)$. If $\overline{B}_U^j(\mathcal{X}_{i'}^{\pi'}) < \overline{B}_U^j(\mathcal{X}_{i'-1}^{\pi'})$, then the policy π'' with $\mathcal{X}_{i'}^{\pi''} = \mathcal{X}_{i'-1}^{\pi'}$ performs strictly better, and thus π' is not optimal. Similarly, if $\overline{B}_U^j(\mathcal{X}_{i'}^{\pi'}) > \overline{B}_U^j(\mathcal{X}_{i'-1}^{\pi'})$, then policy π'' with $\mathcal{X}_{i'-1}^{\pi''} = \mathcal{X}_{i'}^{\pi'}$ performs strictly better, and thus π' is not optimal. If $\overline{B}_U^j(\mathcal{X}_{i'}^{\pi'}) = \overline{B}_U^j(\mathcal{X}_{i'-1}^{\pi'})$, then policy π'' with $\mathcal{X}_{i'}^{\pi''} = \mathcal{X}_{i'-1}^{\pi'}$ would perform at least as good as π'. Thus, there is an optimal policy π^* such that $t_i^{\pi^*} = t_i$. ■

We can hence choose the decision instants such that $t_i^{\pi} = t_i$, and formulate the optimal allocation problem as a finite horizon dynamic decision problem in which the decisions are taken at the beginning of every stationary interval. The Bellman equation for the decision at the beginning of the i^{th} interval, at time t_i, $1 \leq i \leq I$, can be formulated as

$$U^{\pi^*}([t_i, t_{I+1}], \mathcal{X}_{i-1}) = \max_{\mathcal{X}_i} \{\overline{\Gamma}_s(i) - \Gamma_d(A_i) + U^{\pi^*}([t_{i+1}, t_{I+1}], \mathcal{X}_i)\}. \quad (14)$$

The standard way to solve the Bellman equation is via backward induction, but since the predicted demands are known, we can provide an alternative solution.

Theorem 1: Denote by $\Delta_i = t_{i+1} - t_i$ the length of interval i. Then every solution of the following Mixed Integer Linear Programming (MILP) problem:

$$\max \sum_{i=1}^{I} \left\{ \Delta_i \left(\sum_{f\in\mathcal{F}} \overline{B}_f^i x_{i,f} - s_i \right) - \sum_{f\in\mathcal{F}} L_f b_{i,f} \right\} \quad (15)$$

s.t.

$$\sum_{f\in\mathcal{F}} \overline{B}_f^i x_{i,f} - s_i \leq U, \quad \forall 1 \leq i \leq I \quad (16)$$

$$x_{i,f} - x_{i-1,f} - b_{i,f} \leq 0, \quad \forall 1 \leq i \leq I, f \in \mathcal{F} \quad (17)$$

$$\sum_{f\in\mathcal{F}} L_f x_{i,f} \leq S, \quad \forall 1 \leq i \leq I \quad (18)$$

$$b_{i,f} \geq 0, \; x_{i,f} \in \{0,1\}, \quad \forall 1 \leq i \leq I, f \in \mathcal{F} \quad (19)$$

$$s_i \geq 0, \quad \forall 1 \leq i \leq I, \quad (20)$$

is an optimal policy π^* for (14).

Proof: The decision variables $x_{i,f}$ correspond to $f \in \mathcal{X}_i$. The auxiliary decision variables s_i in the bandwidth constraint (16) are the spillover bandwidth in interval i and are used to subtract the spillover cloud traffic from the objective function (15). The auxiliary variable $b_{i,f}$ in content replication constraint (17) is used to include in the objective function (15) the traffic due to storing file f on the dedicated server in interval i if it was not stored there in interval $i-1$. Thus, maximizing (15) under constraints (16) and (17) corresponds to maximizing the traffic cost savings (originally defined in equation (7) in discrete-time domain. Constraint (18) ensures that only feasible file allocations are considered at each time step when solving for the optimal allocation policy. ■

The problem contains $|\mathcal{F}|I$ binary decision variables and $(|\mathcal{F}|+1)I$ continuous decision variables, which allows the MILP to be solved for hundreds of thousands of files and several tens of intervals using state-of-the-art optimization tools.

As the following lemma shows, computational complexity can be reduced by not changing the set of allocated files upon certain intervals without jeopardizing optimality.

Lemma 1: Let $\mathcal{X}^{\pi^*} = (\mathcal{X}_1^{\pi^*}, \ldots, \mathcal{X}_I^{\pi^*})$ be a solution to the optimization problem in Theorem 1. If an allocation $\mathcal{X}_i^{\pi^*}$ for an interval $0 \leq i < I-1$ is such that $\overline{B}_{\mathcal{X}_i}^{i+1} = \sum_{f\in\mathcal{X}_i} \overline{B}_f^{i+1} \geq U$, then there is a solution $\mathcal{X}^{\pi^{*\prime}}$ that differs from $\mathcal{X}^{\pi^*}$ only in that $A_{i+1}^{\pi^{*\prime}} = \emptyset$ and $A_{i+2}^{\pi^{*\prime}} = A_{i+1}^{\pi^*} \cup A_{i+2}^{\pi^*}$, that is, $\mathcal{X}_{i+1}^{\pi^{*\prime}} = \mathcal{X}_i^{\pi^*}$.

Proof: The total cloud traffic induced by the decisions over intervals $i+1$ and $i+2$ under policy $\pi^{*\prime}$ satisfies

$$\Gamma_d(A_{i+1}^{\pi^*}) + \Gamma_d(A_{i+2}^{\pi^*}) \leq \Gamma_d(A_{i+1}^{\pi^{*\prime}}) + \Gamma_d(A_{i+2}^{\pi^{*\prime}}).$$

Furthermore, since by assumption $\overline{B}_{\mathcal{X}_i}^{i+1} \geq U$, the traffic savings is not negatively affected, i.e., $\overline{\Gamma}_s^{\pi^{*\prime}}(i+1) \leq \overline{\Gamma}_s^{\pi^*}(i+1)$ and $\overline{\Gamma}_s^{\pi^{*\prime}}(i+2) \leq \overline{\Gamma}_s^{\pi^*}(i+2)$. ■

Corollary 1: If an allocation $\mathcal{X}_i^{\pi^*}$ for an interval $0 \leq i < I-j$ and some $j > 0$ is such that $\overline{B}_{\mathcal{X}_i}^{i'} \geq U$ for every $i < i' \leq i+j$, then there is a solution $\mathcal{X}^{\pi^{*\prime}}$ that differs from $\mathcal{X}^{\pi^*}$ only in that $A_{i'}^{\pi^{*\prime}} = \emptyset$ and $A_{i+j+1}^{\pi^{*\prime}} = \cup_{i'=i+1}^{i+j+1} A_{i'}^{\pi^*}$. Consequently, $\mathcal{X}_{i'}^{\pi^{*\prime}} = \mathcal{X}_i^{\pi^*}$.

As a consequence, an (optimal) on-line algorithm with a perfect prediction of the future demands for some period I would only need to update the allocation of files in the storage when it reaches a time slot when the current allocation would no longer be able to fully utilize the bandwidth U. Even with this optimization, the computational complexity makes the solution of the MILP infeasible for millions of files and hundreds of intervals. In the following, we therefore consider two approximate solutions.

B. No Download Cost (NDC) Policy

Given the current set $\mathcal{X}_{i-1}$ of stored files, the *NDC* policy considers only the bandwidth demands during the subsequent time interval to perform the maximization. That is, at every decision instance *NDC* solves

$$\mathcal{X}_i^{NDC} = \arg\max_{\mathcal{X}_i} \overline{\Gamma}_s^{\pi}(i). \quad (21)$$

A solution to this maximization problem can be obtained by solving a $0-1$ knapsack problem in which the value of every file is $\overline{B}_f^i \times (t_{i+1} - t_i)$, and the value of the knapsack is at most $U \times (t_{i+1} - t_i)$. Since the weights L_f are integers, the solution can be obtained in $O(|\mathcal{F}|S)$ time using dynamic programming. The *NDC* policy can perform arbitrarily bad, however.

Proposition 2: The approximation ratio $\frac{J^{NDC}}{J^{\pi^*}}$ of the NDC *policy is unbounded.*

Proof: Consider a dedicated server with storage $S = 1$ and bandwidth $U = 1$, two files $\mathcal{F} = \{1,2\}$, initial state $\mathcal{X}_0 = \{1\}$, and let $t_{i+1} - t_i = 1$, $i = 0, \ldots$. The expected bandwidth demands $\overline{B}_f^i$ are ε and 2ε for files 1 and 2, respectively, for pair numbered intervals, and vice versa for odd numbered intervals,

for some $0<\varepsilon<0.5$. The optimal solution is to keep file 1 in the storage, in which case the average cost per interval is 1.5ε. The *NDC* policy is to insert the file with higher bandwidth demand in the storage, in which case the average cost per interval is $1+\varepsilon$. Thus, the approximation ratio is $\frac{J^{NDC}}{J^{\pi^*}}=\frac{1+\varepsilon}{1.5\varepsilon}$, and $\lim_{\varepsilon\to 0}\frac{J^{NDC}}{J^{\pi^*}}=\infty$. ∎

Proposition 3: The approximation ratio of NDC is $\frac{J^{NDC}}{J^{\pi^*}}\leq 1+IS/J^{\pi^*}$.

Proof: In iteration i *NDC* allocates the files $\mathcal{X}$ that maximize $\overline{B}^i_U(\mathcal{X})=\min\left(U,\overline{B}^i_{\mathcal{X}}\right)$, thus $\overline{B}^i_U(\mathcal{X}^i_{NDC})\geq\overline{B}^i_U(\mathcal{X}_{\pi^*})$. In the worst case, *NDC* replaces every file upon every decision unnecessarily, and thus the average per-interval cost of the *NDC* policy is within S of the optimal cost. ∎

Thus, if the amount of data that can be served from the dedicated servers during an interval is significantly higher than the amount of dedicated storage, i.e., $\overline{B}^i_U(\mathcal{X}^i_{NDC})\times E[t_{i+1}-t_i]\gg S$, then the *NDC* policy can be close to optimal.

C. k-Step Look Ahead (k-SLA) Policy

The k-SLA approximation uses a receding horizon of $k>0$ intervals [15]. At the beginning of interval i it solves the MILP (15) to (20) for intervals $[i,\ldots,i+k-1]$ given the initial state $\mathcal{X}_{i-1}$. The set of allocated files in interval i becomes $f\in\mathcal{X}_i \iff x_{1,f}=1$, and the MILP is solved again at the beginning of the subsequent interval.

It is interesting to consider the case of $k=1$. Given the set $\mathcal{X}_{i-1}$ of stored files, the 1-*SLA* policy considers the bandwidth demands and the cost of the allocation during the subsequent time interval i for the maximization. That is, at every decision instance 1-*SLA* solves

$$\mathcal{X}_i^{1-SLA}=\arg\max_{\mathcal{X}_i}\left\{\overline{\Gamma}^{\pi}_s(i)-\Gamma^{\pi}_d(A_i)\right\}. \tag{22}$$

As in the case of *NDC*, if $\sum_{f\in\mathcal{X}_{i-1}}\overline{B}^i_f\geq U$ then $\mathcal{X}_i^{1-SLA}=\mathcal{X}_{i-1}$ is 1-*SLA* optimal, and thus $A_i^{1-SLA}=\emptyset$. Unfortunately, the 1-*SLA* policy is, similar to *NDC*, suboptimal even for very simple problems, as the following shows.

Proposition 4: The approximation ratio $\frac{J^{1-SLA}}{J^{\pi^*}}$ *of the* 1-SLA *policy is unbounded.*

Proof: To prove the proposition, we construct an example with unbounded approximation ratio. Consider a dedicated server with storage $S=1$ and bandwidth $U=1$, two files $\mathcal{F}=\{1,2\}$, initial state $\mathcal{X}_0=\{1\}$, and let $t_{i+1}-t_i=1$, $i=1,\ldots$. The expected bandwidth demands $\overline{B}^i_f$ are 2ε and $1+\varepsilon$ for file 1 and for file 2, respectively, for every interval i for some $0<\varepsilon<0.5$. The optimal solution is $\mathcal{X}_i=\{2\}$, $i\geq 1$, in which case the average cost per interval is 3ε. The 1-*SLA* policy is to keep file 1 in the storage, in which case the average cost per interval is $1+\varepsilon$. Thus, the approximation ratio is $\frac{J^{1-SLA}}{J^{\pi^*}}=\frac{1+\varepsilon}{3\varepsilon}$, and $\lim_{\varepsilon\to 0}\frac{J^{1-SLA}}{J^{\pi^*}}=\infty$. ∎

Unlike *NDC*, in the worst case, 1-*SLA fails* to replace every file upon every decision. It fails to replace a file f only if the per interval cost of the file is within L_f of its long term average cost. This observation allows us to obtain a bound on the approximation ratio of *k-SLA* if the average demands are bounded.

Proposition 5: Consider a system in which the average demand of each file inserted into the dedicated storage by an optimal policy π^* *is lower bounded by a factor* $\rho>0$ *such that the demand of each such file satisfies* $\rho\frac{1}{I}\sum_{i=0}^{I}\overline{B}^i_f\Delta_i\geq L_f$. *Then, for* $k>\frac{\rho I}{I-\rho}$ *the approximation ratio of* k-SLA *is*

$$\frac{J^{k-SLA}}{J^{\pi^*}}\leq\frac{1}{1-\frac{\rho}{k}(1+\frac{k}{I})}. \tag{23}$$

For $I\to\infty$ *the approximation ratio is bounded by a geometric series with ratio* ρ/k.

Proof: Consider an initial allocation $\mathcal{X}_0$, and let $\mathcal{X}^{\pi^*}=(\mathcal{X}_0^{\pi^*},\ldots)$ be the allocation under an optimal policy π^*. The worst case appproximation ratio of *k-SLA* is achieved in a scenario (i) when π^* involves replacing all files in storage in the first interval, after which it does not change the set of allocated files, (ii) there is no spillover traffic, and (iii) *k-SLA* always allocates the set of files complementary to $\mathcal{X}^{\pi^*}$, i.e., it fails to introduce the same files as the optimal policy. The cost under the optimal policy π^* for such a scenario is

$$J^{\pi^*}=\sum_{i=1}^{I}\sum_{f\notin\mathcal{X}_i^{\pi^*}}\overline{B}^i_f\Delta_i+\sum_{f\in\mathcal{X}_i^{\pi^*}}L_f\geq\sum_{i=1}^{I}\sum_{f\notin\mathcal{X}_i^{\pi^*}}\overline{B}^i_f\Delta_i, \tag{24}$$

because $\mathcal{X}_i^{\pi^*}\cap\mathcal{X}_0^{\pi^*}=\emptyset$. Consider now the cost under *k-SLA*, which fails to introduce the files $\mathcal{X}_i^{\pi^*}$. By the definition of *k-SLA* this happens if for every $0<i_0\leq I-k$

$$\sum_{i=i_0}^{i_0+k-1}\sum_{f\in\mathcal{X}_i^{\pi^*}}\overline{B}^i_f\Delta_i-\sum_{f\in\mathcal{X}_i^{\pi^*}}L_f\leq\sum_{i=i_0}^{i_0+k-1}\sum_{f\notin\mathcal{X}_i^{\pi^*}}\overline{B}^i_f\Delta_i. \tag{25}$$

Using the above expressions we can bound the cost for *k-SLA*

$$\begin{aligned}
J^{k-SLA} &= \sum_{i=1}^{I}\sum_{f\in\mathcal{X}_i^{\pi^*}}\overline{B}^i_f\Delta_i && (26)\\
&\leq \sum_{i=1}^{I}\sum_{f\notin\mathcal{X}_i^{\pi^*}}\overline{B}^i_f\Delta_i+\left\lceil\frac{I}{k}\right\rceil\sum_{f\in\mathcal{X}_i^{\pi^*}}L_f && (27)\\
&< \sum_{i=1}^{I}\sum_{f\notin\mathcal{X}_i^{\pi^*}}\overline{B}^i_f\Delta_i+\frac{I+k}{k}\rho\frac{1}{I}\sum_{i=0}^{I}\sum_{f\in\mathcal{X}_i^{\pi^*}}\overline{B}^i_f\Delta_i && (28)\\
&\leq J^{\pi^*}+\frac{I+k}{k}\frac{\rho}{I}J^{k-SLA}. && (29)
\end{aligned}$$

Rearranging and solving for the ratio J^{k-SLA}/J^{π^*} completes the proof of the proposition. ∎

Consequently, if $k>>\rho$ then *k-SLA* is close to optimal. Furthermore, if the amount of data that can be served from the dedicated servers during an interval is high (i.e., ρ is low), then *k-SLA* is close to optimal for low values of k.

V. Performance Evaluation

A. Synthetic trace-based evaluation

We first evaluate the proposed algorithms on synthetic traces motivated by the measured traces used in Section V-B. Each

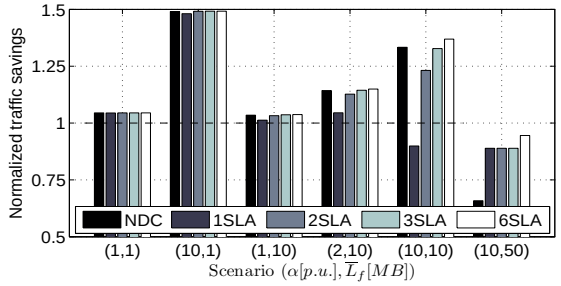

Fig. 3. Normalized traffic savings for various $(\alpha, \overline{L}_f)$ scenarios. *NDC* performs well for low average file size ($\overline{L}_f = 1MB$), when gains are highest. *k-SLA* performs as good for high *k* irrespective of the file size.

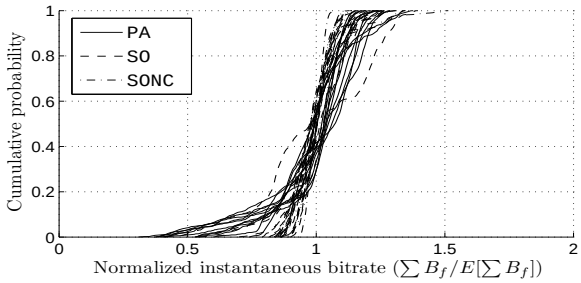

Fig. 4. CDF of the instantaneous aggregate demand of approximately 2000 allocated files normalized by the average aggregate demand during 12 stationary intervals of 30 minutes each, for *PA*, *SO* and *SONC* scenarios.

synthetic trace is one week-long, and consists of 3 sets of 1000 files each. The bandwidth demand of each file follows a sinusoidal with a daily period; the minimum/maximum ratio over a day is normally distributed with mean and standard deviation of 0.075 and 0.075, respectively. The average time-of-day peak times of the 3 file sets are offset by on average 8 hours. Within each file set, the time-of-day peak times are normally distributed with a standard deviation of 2 hours.

The file sizes are distributed uniformly with mean $\overline{L}_f$ on $[\overline{L}_f/2, 3\overline{L}_f/2]$. The peak bandwidth demand of every file was drawn from a bounded Pareto distribution with lower bound B^{min}_{peak}, upper bound $B^{max}_{peak} = 2 \cdot 10^3$ KB/sec, and shape parameter α. For given α we chose B^{min}_{peak} such that the Pareto distributions have the same average of 2.5 KB/sec per file.

Fig. 3 shows the traffic savings of *NDC* and *k-SLA* normalized by the traffic savings under the policy that allocates the files that have the highest average bandwidth demand over the entire period for six shape parameter α and file size $\overline{L}_f$ combinations, $U = 10$Mbps and $S = 100\overline{L}_f$. We used three different shape parameter-lower bound combinations. First, $\alpha = 1.01$, $B^{min}_{peak} = 300$, which results in a Zipf-like rank-popularity plot with tail exponent $\gamma = 1$, as the tail exponent γ of the Zipf distribution is $\gamma = 1/\alpha$. Second, $\alpha = 10$, $B^{min}_{peak} = 2,299$, which results in almost uniformly distributed demands. Third, $\alpha = 2$, $B^{min}_{peak} = 1,278$, which resembles the rank-popularity of the measured traces discussed in Section V-B.

The results show that dynamic allocation can outperform the static (non-causal) allocation by up to 50%. Highest gains are achieved for high shape parameter α (nearly uniform demand distribution) and low average file sizes. *NDC* performs comparable to *k-SLA* for low average file sizes (i.e., when moving files to storage involves little penalty), but it fails otherwise. It is noteworthy that albeit *k-SLA* is computatationally more intensive, for sufficiently large *k* it performs consistently better than *NDC* for all average file sizes.

B. Measured trace-based evaluation

For the second evaluation, we use a trace collected from a commercial audio on-demand streaming system called Spotify [16], [17]. The system has over 24 million active (in the last month) users in 28 countries, among them the U.S., and a catalogue of over 20 million tracks, with new tracks added continuously. The trace we use was collected during a week in March 2011, and contains the bandwidth demands from a single country for 1 million tracks chosen uniformly at random. Tracks are encoded at 160kbps and/or 320kbps and the average track size is approximately 5MB.

Clients obtain the data for a track, in order of preference, from the local cache, from other clients' caches, and from the content server. The trace allows us to distinguish between the data requested by each client from the three sources, we can thus create three demand scenarios: the actual demands at the servers in the peer-assisted system (PA), the demands including data downloaded from other clients which resembles a server-only system (SO), and the demands including data downloaded from other clients and the local cache which resembles a server-only system with no cache (SONC).

1) Mean-value and Rank-parameter Validation: Fig. 4 shows the instantaneous aggregate bandwidth demand $\sum_{\mathcal{X}_i} B^i_f(t)$ of allocated files normalized by the average aggregate demand $\overline{B}^i_{\mathcal{X}_i}$ during 12 stationary intervals of 30 minutes each, for the *PA*, *SO* and the *SONC* scenarios. The set of files is obtained using *NDC* with 8 GB of storage and 30Mbps unmetered bandwidth for each interval, and thus $|\mathcal{X}_i| \approx 1000$ in each interval. The instantaneous aggregate bandwidth demand is within 20% of the average for about 80% of the intervals for all scenarios and intervals, but for one outlier. This shows that the mean value approximation over relatively short intervals is reasonable despite the large (an order of magnitude) diurnal fluctuations of the aggregate bandwidth demands.

Fig. 5 shows the ranked average bandwidth demands of the tracks calculated for 12 intervals of 30 minutes each, for the *PA*, *SO* and the *SONC* scenarios, that is, in total 36 curves. The curves are normalized by the highest average demand overall. We observe three important characteristics. First, for a particular scenario the rank-demand curve for each interval follows Zipf's law but with slightly different shape parameters. Second, the bandwidth demand of the most popular file for a particular scenario can differ by approximately an order of magnitude between the different intervals. Third, the bandwidth demands between intervals differ most under the *SONC* scenario, while they differ least under the *PA* scenario.

We fitted a Zipf curve to the rank statistics of the average bandwidth demand in every 30 minutes long interval to estimate the exponent α of the Zipf curve. Fig. 6 shows the Zipf exponent α averaged for the same 30 minutes long interval of every day for the *PA*, *SO* and the *SONC* scenarios, and their 95% confidence intervals. It is important to note that the confidence intervals are small, which shows that the Zipf

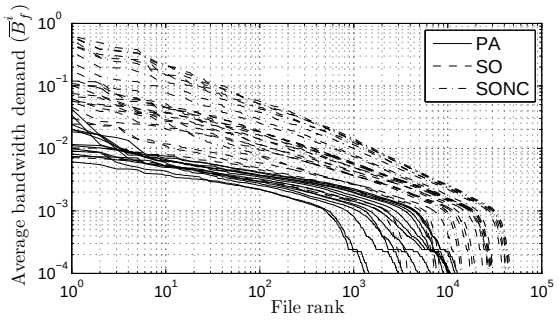

Fig. 5. Average bandwidth demand rank statistics of the tracks during 12 stationary intervals of 30 minutes each, for *PA*, *SO* and *SONC* scenarios.

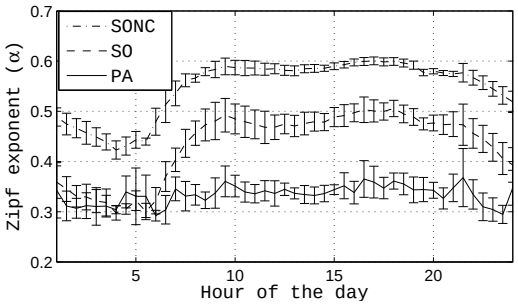

Fig. 6. Average Zipf exponent (α) as a function of the hour of the day for *PA*, *SO* and *SONC* scenarios.

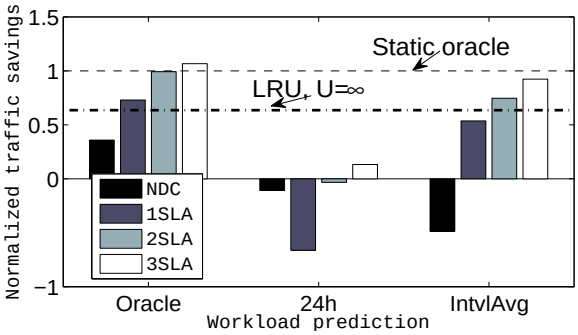

Fig. 7. Normalized traffic savings for 4 allocation policies under 3 prediction schemes for the *PA* scenario, $S = 8$GB, $U = 30$Mbps.

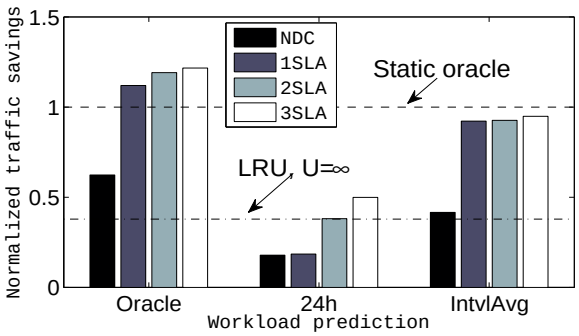

Fig. 8. Normalized traffic savings for 4 allocation policies under 3 prediction schemes for the *PA* scenario, $S = 1$GB, $U = 30$Mbps.

exponents are similar at the same hour of the day over different days. The figure also shows that the Zipf exponent depends heavily on the hour of the day for the *SO* and for the *SONC* scenarios. While there have been several measurement studies that showed that the popularity of content workloads tends to follow Zipf's law [8], [18], we are not aware of any work that shows that the popularity-rank distribution of contents evolves with time according to a periodic pattern. The Zipf exponent also depends on the scenario. Under the *PA* scenario, which corresponds to the peer-assisted system with cache, the exponent is fairly steady and low, because the most popular files are typically downloaded from peers rather than from the server. This means, however, that the actual server bandwidth demand distribution is closer to uniform under the *PA* scenario than under the *SO* and the *SONC* scenarios, which based on the results in Section V-A makes dynamic allocation for the *PA* scenario more beneficial.

2) Gain of Dynamic Content Allocation: We consider three predictions of the average bandwidth demands in order to investigate the sensitivity of the policies to the accuracy of the bandwidth demand predictions and to the average demand per file. The *oracle* prediction is the actual average bandwidth demand, and is thus the case of perfect prediction. The *24h* prediction uses $\overline{B}_f^i = \overline{B}_f^{i-48}$, that is, the average bandwidth demand 24 hours before the actual interval. The *IntvlAvg* prediction uses the weekly average demand of the intervals at the same hour of the day, except for the interval to be predicted, that is, $\overline{B}_f^i = \frac{1}{6}\sum_{j\%48=i\%48, j\neq i} \overline{B}_f^j$. As a baseline we use the policy that allocates the files that have the highest average bandwidth demand over the entire period, i.e., a *static oracle allocation*. We also simulated an LRU cache with and without ($U = \infty$) bandwidth limit.

Fig. 7 shows the traffic savings for a configuration with $S = 8$GB and $U = 30$Mbps normalized by the traffic savings under the static oracle allocation. The figure shows that *NDC* performs poorly even for the *oracle* prediction. Although the actual amount of data served from the server is highest for *NDC*, it changes the allocation too frequently, which leads to many files being downloaded from the cloud to the servers. *k-SLA* outperforms the static allocation for a horizon of $k = 3$ for the *oracle* prediction, and performs almost as well for the *IntvlAvg* prediction. This shows that (i) the *IntvlAvg* prediction is rather accurate, and (ii) with a good prediction of the average bandwidth demands *k-SLA* can actually lead to higher cost saving than the static allocation. This is in contrast to the poor performance observed using the *24h* prediction, which is too noisy. A comparison of the results for *k-SLA* with different horizon lengths also shows that a horizon of $k = 3$ gives little benefit over a horizon of $k = 2$. The result for *LRU* was -6.07 (not shown), which shows that caching is far worse than the simplest dynamic allocation policy because (i) it does not account for spillover traffic and (ii) moves rarely accessed files to cache. Even *LRU-*∞ performs poorly compared to *k-SLA*.

Fig. 8 shows corresponding results for storage $S = 1$GB. Since the dedicated storage is smaller, the average bandwidth demand per file is higher, and thus both *NDC* and *k-SLA* perform significantly better than for $S = 8$GB. The relative savings for *LRU* (not shown) was below -6 due to spillover and excessive evictions, and even *LRU* without the bandwidth limit performs almost as bad as the simplest allocation policy. This confirms that if the aggregate bandwidth demand per file is high compared to the storage size, the approximate dynamic allocation policies perform well. For low average aggregate bandwidth demand the cloud traffic due to allocating a file to

storage cancels most of the gains, and thus a static allocation is close to optimal. Importantly, in both scenarios the allocation policies by far outperform traditional caching.

VI. Related Work

Frameworks for cloud-assisted P2P streaming to accommodate time varying demands have been considered for both live streaming [19] and VoD [20]. In VoD and live streaming systems chunk delivery needs to be approximately in-order, unlike in the case of the content distribution systems we address in this paper, where the delivery order is unrestricted. Apart from this difference, the consideration of a hybrid cloud system with both dedicated/unmetered and elastic bandwidth costs distinguish our work from the above papers. We show that by leveraging the benefits of both types, a content provider can significantly lower its delivery cost, compared to only using one or the other, and then evaluate candidate policies for how to best use these resources.

Niu et al. [21] makes a case for bigger content providers negotiating reservations for bandwidth guarantees from the cloud to support continuous media streaming. Within this context they argue that it is beneficial to multiplex such bandwidth reservations, and show that (within their framework) the market would have a unique Nash equilibrium, with the bandwidth reservation price critically depending on the market demand. In related work, the same authors present a social welfare optimization formulation, for which they present distributed solution methods [6]. Qui et al. [22] present an optimization framework for the migration process of a content provider moving its content distribution to the cloud. In contrast to these works, our focus is on how to allocate and best utilize the available server and cloud resources. We build our model based on two complementing and existing pricing and service models; one on-demand and one fixed. Recent work has considered prediction methods for on-demand service workloads (e.g., [4], [5], [6]). Our work is orthogonal to these works, as our focus is on the performance of the allocation policies, and the performance of all prediction methods is bounded by the cost savings of the oracle policy we consider.

Finally, the MILP formulation can be considered a variant of the 0-1 Knapsack problem [23]. In contrast to the traditional multiple-knapsack problem with identical capacities (MKP-I), one knapsack for each time interval [23], we allow for the same file to be present in consecutive intervals and introduce a penalty for each time something is introduced in the knapsack that was not present in the previous knapsack, as well as a cap on the maximum possible profit. We are not aware of any treatment of such a coupled multiple-knapsack problem.

VII. Conclusion

We considered the problem of dynamic content allocation for a hybrid content distribution system that combines cloud-based storage with dedicated servers and upload bandwidth. We formulated the problem of allocating contents to the dedicated storage as a finite horizon dynamic decision problem, provided the exact solution to the discrete time approximation as a MILP, and provided computationally efficient approximations with provable approximation ratios. Using traces from a commercial content distribution system we showed that when upload bandwidth is abundant (high U/S ratio), the simple NDC approximation works well, but a look-ahead policy is needed otherwise. Dynamic allocation can provide up to 50% gain compared to static allocation, and outperforms LRU caching by far.

VIII. Acknowledgements

The authors thank R. Srikant for comments on the draft of this work. Dán was partially supported by grant 2010-5812 of the Swedish Research Council and Carlsson by Ceniit.

References

[1] Bertran et al., "RFC6770: Use cases for content delivery network interconnection," Nov. 2012. [Online]. Available: http://tools.ietf.org/html/rfc6770
[2] N. Megiddo and D. S. Modha, "Arc: A self-tuning, low overhead replacement cache," in *Proc. USENIX FAST*, 2003.
[3] G. Szabo and B. Hubermann, "Predicting the popularity of online content," *Communications of the ACM*, vol. 53, no. 8, pp. 80–88, 2010.
[4] D. Niu, Z. Liu, B. Li, and S. Zhao, "Demand forecast and performance prediction in peer-assisted on-demand streaming systems," in *Proc. IEEE INFOCOM Mini-Conference*, 2011.
[5] G. Gusun, M. Crovella, and I. Matta, "Describing and forecasting video access patterns," in *Proc. IEEE INFOCOM Mini-Conference*, 2011.
[6] D. Niu, C. Feng, and B. Li, "Pricing cloud bandwidth reservations under demand uncertainty," in *Proc. ACM SIGMETRICS/Performance*, 2012.
[7] D. K. Krishnappa, S. Khemmarat, L. Gao, and M. Zink, "On the feasibility of prefetching and caching for online tv services: A measurement study on hulu," in *Proc. PAM*, 2011.
[8] K. Sripanidkulchai, B. Maggs, and H. Zhang, "An analysis of live streaming workloads on the internet," in *Proc. ACM IMC*, 2004.
[9] G. Dán and N. Carlsson, "Centralized and distributed protocols for tracker-based dynamic swarm management," *IEEE/ACM Trans. on Networking*, vol. 21, no. 1, pp. 297–310, 2013.
[10] R. Michel, "On berry-esseen results for the compound poisson distribution," *Insurance: Math. and Econ.*, vol. 13, no. 1, pp. 35–37, 1993.
[11] A. J. E. M. Janssen, J. S. H. van Leeuwaarden, and B. Zwart, "Gaussian expansions and bounds for the poisson distribution applied to the erlang b formula," *Adv. in Appl. Probab.*, vol. 40, no. 1, pp. 122–143, 2008.
[12] N. Carlsson, D. L. Eager, and A. Mahanti, "Using Torrent Inflation to Efficiently Serve the Long Tail in Peer-assisted Content Delivery Systems," in *Proc. IFIP/TC6 Networking*, May 2010.
[13] N. Carlsson, G. Dan, D. Eager, and A. Mahanti, "Tradeoffs in cloud and peer-assisted content delivery systems," in *Proc. IEEE P2P*, 2012.
[14] A. Siegel, "Toward a usable theory of chernoff bounds for heterogeneous and partially dependent random variables," New York, NY, USA, Tech. Rep., 1995.
[15] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.
[16] Spotify, http://www.spotify.com.
[17] G. Kreitz and F. Niemelä, "Spotify - large scale, low latency, P2P music-on-demand streaming," in *Proc. IEEE P2P*, 2010.
[18] G. Dán and N. Carlsson, "Power-law revisited: A large scale measurement study of P2P content popularity," in *Proc. IPTPS*, Apr. 2010.
[19] F. Wang, J. Liu, and M. Chen, "CALMS: Cloud-Assisted Live Media Streaming for Globalized Demands with Time/Region Diversities," in *Proc. IEEE INFOCOM*, 2012.
[20] D. Niu, H. Xu, B. Li, and S. Zhao, "Quality-Assured Cloud Bandwidth Auto-Scaling for Video-on-Demand Applications," in *Proc. IEEE INFOCOM*, 2012.
[21] D. Niu, C. Feng, and B. Li, "A theory of cloud bandwidth pricing for video-on-demand providers," in *Proc. IEEE INFOCOM*, 2012.
[22] X. Qiu, H. Li, C. Wu, Z. Li, and F. C. Lau, "Cost-minimizing dynamic migration of content distribution services into hybrid clouds," in *Proc. IEEE INFOCOM Mini-Conference*, 2012.
[23] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Springer, 2004.