A Hybrid Algorithm for Content Placement in Distributed Video on Demand Systems

James Y. Yang
Department of ECE, Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 W Main Street, Urbana, IL 61801
Email: jyyang3@illinois.edu

Bruce Hajek
Department of ECE, Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 W Main Street, Urbana, IL 61801
Email: b-hajek@uiuc.edu

Abstract—We study the content placement problem for cache delivery video-on-demand systems under static random network topologies with fixed heavy-tailed video demand. The performance measure is the amount of server load; we wish to minimize the total download rate for all users from the server and maximize the rate from caches. Our approach reduces the analysis for multiple videos to consideration of decoupled systems with a single video each. For each placement policy, insights gained from the single video analysis carry back to the original multiple video content placement problem. Finally, we introduce a hybrid placement technique that achieves near optimal performance with low complexity.

I. INTRODUCTION

A video-on-demand (VoD) system is an online video delivery system in which peers can request which videos to watch. In order for peers to watch videos without interruptions or large delays, the system must meet stringent delivery requirements - peers need to begin to download quickly and stream at the playback rate.

In our analysis, we primarily study the content placement problem: Given a set of caches, what set of videos should be stored at each cache under storage constraints, demand, and network topology. There are two types of storage methods for this problem: 1. Whole storage - videos are stored as whole copies, and 2. Fractional storage - with the help of source codes such as maximum-distance-separable (MDS) codes, videos are coded and stored as fractions of a copy. When a peer requests a video, the whole storage architecture requires the peer to download only from one of its connected caches containing the video. In the fractional storage architecture, a peer can simultaneously download from multiple connected caches containing the coded fractions of the video, and the download rates are additive by the property of the MDS code.

Under these two storage methods, placement policies can be categorized into adaptive or fixed (non-adaptive) placement. Under adaptive placement policies, copies of a particular video are stored in the set of caches with the most received requests. In our model, we assume videos requested by peers are known before the placement decisions are made. To achieve this assumption, a recommender system could be used for predicting peer requests or high speed links from server to caches could allow a sequence of video chunks to be placed immediately after peer requests. In addition, this assumption yields a performance bound. Under fixed placement policies, caches are oblivious to the current demand so random placement of copies of a particular video has the same expected performance as deterministic placement of the copies.

Adaptive fractional storage placement minimizes the server load and can be implemented in a distributed way. However, it requires overhead and computational power to encode and decode the videos. Optimal adaptive whole storage placement is combinatorial in nature and becomes computationally intractable when the system grows large. The fixed fractional and fixed whole storage placement policies are simple to implement, but for videos that are not very popular, they do not serve the demand well using a small number of copies. The fraction of demand satisfied by fixed fractional policies grows linearly with the number of copies placed, and dominates the expected demand satisfied by the fixed whole policies. In this paper we introduce a hybrid placement policy which nearly achieves the minimum server load, but with less complexity than the adaptive fractional storage placement policy.

II. RELATED WORK AND CONTRIBUTION

VoD cache delivery systems have received wide attention for their benefits - reduction of content delivery cost and improvement of the end-user performance. Popular websites such as Youtube have been aggressively deploying cache servers of widely varying sizes at many locations around the world [1]. In addition, cache servers in cache delivery systems appear as television set top boxes or personal computers in peer-to-peer networks like PPLive [2]. There are a number of works on content placement [3]-[11]. Almeida et al. [3] studied content placement and routing optimization in cache delivery systems subject to path delivery cost under a fixed topology. Boufkhad et al. [4] derived bounds on the number of videos that can be served in the system subject to storage constraint, upload constraint, and cache connection degree. Tan et al. [5] established an asymptotically optimal content placement strategy subject to a storage constraint and loss network model. Zhou et al. [6] focused on balancing the workload of caches. Laoutaris et al. [7] studied cache storage resource allocation. Wu and Li [8] studied the optimal cache replacement algorithm and suggested that the simplest heuristics perform as well as the optimal algorithms, with very insignificant differences.
Applegate et al. [9] formulated the adaptive whole storage placement problem as a mixed integer program (MIP) subject to a storage constraint and link bandwidth constraint. The adaptive whole storage placement problem is solved approximately by MIP. Zhang [10] used MDS codes to relax the integer constraint and converted the integer program into a convex, adaptive fractional storage placement problem that can be solved exactly. This result provides an upper bound on the performance of any content placement policy, including all single video placement policies. Golrezaei et al. [11] proposed a similar model which also analyzes both whole and fractional storage placement policies. Their model assumes knowledge of video demand statistics but not the actual demand, so their algorithm adapts to the network topology and video demand distribution (Zipf) but not directly to the actual demand.

Our main contribution presented in this work is a hybrid storage multiple video placement policy, which is a suboptimal alternative with low complexity to the optimal placement algorithm adapts to the network topology and video demand statistics but not the actual demand, so their algorithm adapts to the network topology and video demand distribution (Zipf) but not directly to the actual demand.

To focus strictly on the analysis of content placement policies in the VoD cache system, we adapt a simple closed homogeneous system model. We consider the server to be external to the cache system and to provide help only when caches within the system cannot satisfy all the peer demand, as shown in Figure 1. In the model, the numbers of peers, videos, and caches remain fixed. Each peer is connected to an equal number of caches connected to a peer. We assume peer connections (the set of caches connected to a peer) are also uniformly distributed.

We have a model of a random but fixed network topology (a graph of cache-peer connections). Each video in the system occupies the same storage space and has the same playback rate. Peers in the system watch videos continuously and independently select videos according to a heavy-tailed popularity distribution, which we assume is a Zipf distribution. We ignore link bandwidth constraints, imposed by an underlay network.

The notation of the model is summarized as follows:

- \( H \): set of caches
- \( U \): set of peers
- \( M \): set of videos, ordered according to popularity
- \( G \): set of all possible network topology graphs
- \( N_u^g \): set of caches connected to peer \( u \) under graph \( g \in G \)
- \( L = |N_u^g| \): equal number of caches connected by peers
- \( H_m \): set of caches storing video \( m \)
- \( U_m \): set of users requesting video \( m \)
- \( S_h \): storage capacity of cache \( h \) in units of videos
- \( Zipf(K, 0.8) \): probability distribution over \( \{1, ..., K\} \)
- \( p(m) \): video popularity distribution (i.e. probability the video requested by a peer is video \( m \))
- \( x_{hu} \): download rate of peer \( u \) from cache \( h \)
- \( W_{hm} \): fraction of video \( m \) on cache \( h \)
- \( C_m \): total copies of video \( m \) stored in the cache system
- \( \alpha(m) \): probability video \( m \) is present in a typical cache

Given the set of caches \( H \) and the set of peers \( U \), under a fixed network topology \( g \in G \), each peer \( u \) is connected to \( |N_u^g| = L \) caches selected randomly and uniformly. Given the set of videos \( M \), the probability a peer requests video \( m \), \( p(m) \), follows the Zipf\([|M|,0.8]\) distribution, and the mean number of peers requesting video \( m \) satisfies \( E[|U_m|] = |U| \cdot p(m) \). Each cache \( h \) has an equal storage capacity of \( S_h \) units of video and stores \( W_{hm} \) units of video \( m \), where \( W_{hm} \in \{0,1\} \) for whole storage placement and \( W_{hm} \in [0,1] \) for fractional storage placement.

The cost function to be minimized is the server load, or equivalently, the utility function to be maximized is the total download rate for all users from the caches.

\[
\max \sum_{u \in U} \min \left\{ \sum_{h \in H_u} x_{hu}, 1 \right\} \\
\text{w.r.t.} \quad x_{hu} \text{ and } W_{hm} \forall u \in U, h \in H, m \in M \\
\text{s.t.} \quad x_{hu} \leq W_{hm} \forall h, u \text{ and } m : u \in U_m, \\
\quad \sum_{m \in M} W_{hm} \leq S_h \forall h, \\
\quad W_{hm} \in \{0,1\} \text{ or } W_{hm} \in [0,1] \forall u, h, m
\]

IV. SINGLE VIDEO PLACEMENT POLICY ANALYSIS

This section analyzes the four storage method and placement policy pairs for a single video.

Fig. 1. Cache system of three caches, four peers and two random connections per peer
A. Fixed Whole Storage Placement Policy

For fixed whole storage placement policies, cache content is not placed in according to the actual demand and topology \((N^m_u)\) is ignored. Given an integer number of video copies, \(C\), to be placed in the cache system, a set of caches of cardinality \(C\) is randomly selected to store one whole video copy each, and \(|H_m| = C\). A peer can download from at most one of its \(L = |N^m_u|\) connected caches that store the requested video. Let \(p_{miss}(m)\) denote the probability a peer is not connected to a cache storing video \(m\), given by \(p_{miss}(m) = \frac{(|H|^C - C)}{|H|^C}\). Given the set of peers, \(U\), requesting video \(m\), the expected number of peers that are served by the cache system without help from the server is given by:

\[
|U| \cdot (1 - p_{miss}(m))
\]

The following proposition provides an upper bound on \(p_{miss}(m)\) and therefore a lower bound on the expected total download rate for video \(m\) provided by the caches. Proofs of all propositions are provided in the full version of this paper on arXiv:1307.0849.

**Proposition 4.1:** \(p_{miss}(m) \leq (1 - \alpha(m))^L = (1 - \frac{C}{|H|^C})^L\).

B. Fixed Fractional Storage Placement Policy

For fixed fractional storage placement policies, caches’ video catalogs also remain fixed regardless of the demand and topology \((N^m_u)\) is ignored. For the fixed uniform fractional storage placement policy, with a given number (possibly non-integer) of video copies, \(C\), in the cache system, each cache stores the uniform fraction, \(W_h = \frac{C}{|H|^C}\), of video \(m\).

**Proposition 4.2:** Among all fixed fractional storage placements, the uniform fractional storage placement maximizes the expected total download rate.

C. Adaptive Whole Storage Placement Policy

For adaptive whole storage placement policies, caches select their video catalogs in response to the actual demand. Optimal adaptive whole storage placement is an NP-hard problem, but in this context near optimal performance is provided by a greedy algorithm.

Let \(p_{miss}(m)\) denote the probability a peer is not connected to a cache storing video \(m\), given by \(p_{miss}(m) = \frac{(|H|^C - C)}{|H|^C}\). Let \(\alpha(m)\) denote the fraction of peers requesting video \(m\) and stores a copy in that cache. Then, it peels away (removes) all peers connected to that cache requesting video \(m\).

**Algorithm 1** Greedy peeling algorithm

1: while Not all \(C\) copies of video \(m\) are placed in the caches do
2: - Find \(h_{max}\) with the most connected peers requesting video \(m\)
3: - Store a copy of video \(m\) in cache \(h_{max}\)
4: - Remove all peers connected to \(h_{max}\) requesting video \(m\) from the cache system
5: end while

D. Adaptive Fractional Storage Placement Policy

For adaptive fractional storage placement policies, caches’ video catalogs are also selected in response to the actual demand. Given the number (possibly non-integer) of video copies \(C\) to be placed in the cache system, each cache stores some fraction of video \(m\), \(W_h\). The values \((W_h : h \in H, m \in M)\) are chosen to maximize the total download rate provided by the cache system without help from the server. Since adaptive fractional storage placement relaxes the integer constraint in (1), we can solve the convex optimization problem exactly. The cache system’s fractional storage placement, routing, and upload rates converges optimally by the primal-dual algorithm, shown as Algorithm 2, which is a stripped down variation of the algorithm in [10]. The notation \(\lceil a \rceil^\pm\) below denotes \(a\) if \(x_{h,u} > 0\) and \(\max \{0, a\}\) if \(x_{h,u} \leq 0\).

**Algorithm 2** Primal-dual algorithm

Primal 1: update the download rates with \(\delta_{h,u} > 0\)

\(\hat{x}_{h,u} = \left[ \delta_{h,u} \cdot \left( \frac{\sum_{h' \in H} x_{h',u} < 1} - \lambda_{h,u} \right) \right]^+\)

Dual 1: update the price of download rates with \(\kappa_{h,u} > 0\)

\(\hat{\lambda}_{h,u} = \left[ \kappa_{h,u} \cdot (x_{h,u} - W_h) \right]_{\hat{\lambda}_{h,u}^+}\)

Primal 2: update the fraction of the video stored with \(\nu_h > 0\)

\(\hat{W}_h = \left[ \nu_h \cdot \left( \sum_{h \in H} C - \hat{\lambda}_{h,u} \right) - \omega \right]_{\hat{W}_h}^+\)

Dual 2: update the price of cache storage with \(\nu_h > 0\)

\(\hat{\omega} = \left[ \nu_h \cdot \left( \sum_{h \in H} W_h - C \right) \right]_{\hat{\omega}^+}\)

V. SINGLE VIDEO PLACEMENT POLICY COMPARISONS

This section explores the trade-offs between performance and complexity of the placement policies for a single video discussed in section IV. To look for potential patterns in single video placement performances, we choose 20 and 2000 peers representing two different popularity levels, where each peer is connected to four random caches out of the 50 caches selected uniformly. Videos with 20 or fewer peer requests represent the set of videos with below average popularity. A video with 2000 peer requests represents the most demanded video. The performance of single video placement policies derived from the analysis in section IV are shown in Figures 2 and 3. The fraction of total download rate provided by the caches
is plotted versus the number of copies of the video stored in the cache system. The abbreviation FW, FF, AW, and AF represents the fixed whole, fixed fractional, adaptive whole, and adaptive fractional storage placement policy, respectively.

![Graph](image1.png)

**Fig. 2.** Single video performance of a cache system of 50 caches, 20 peers and four random connections per peer

Thus, the adaptive placement of whole copies of a video is imbalanced as few caches have significantly higher demand. The abbreviation FW, FF, AW, and AF is plotted versus the number of copies of the video stored in the cache system. The maximum of the two solid curves is nearly as many peers, but with less overhead. As observed in section V, for videos with low popularity, the adaptive whole storage placement policy serves nearly as many peers as the adaptive fractional storage placement policy. And for popular videos, the fixed fractional policy serves nearly as many peers as the adaptive fractional storage placement policy. This suggests using a hybrid policy that follows the adaptive whole placement policy for videos with low popularity and the fixed fractional placement policy for popular videos.

From the single video analysis, we have obtained the single video service curves for each video \( m \in M \), denoted as \( f_{m,0}(C_m) \) and \( f_{m,1}(C_m) \) for the fixed fractional and adaptive whole storage placement policies, respectively, which are concave functions of the number of copies stored, \( C_m \). For each video \( m \), the hybrid storage placement policy chooses to apply one of the two placement policies, and the number of copies \( C_m \) to store. The choices are made to maximize the total number of peers served subject to the total storage constraint, \( S \). Thus, with \( f_m \triangleq \max \{ f_{m,0}(C_m), f_{m,1}(C_m) \} \), the optimization problem can be stated as follows:

\[
\max \quad U(C) \triangleq \sum_{m \in M} f_m(C_m) \quad (3)
\]

w.r.t. \( C_m, \ m \in M \)

s.t. \( \sum_{m \in M} C_m \leq S \)

However, (3) is not a concave optimization problem, because the functions \( f_m \) are not concave. To address this problem, we use the concave hull of the objective function, to arrive at the a new optimization problem with the following objective function:

\[
\max \quad \overline{U}(C) \triangleq \sum_{m \in M} \overline{f}_m(C_m) \quad (4)
\]

where for each video \( m \), \( \overline{f}_m \) denotes the concave hull of \( f_m \).

Let \( V^* \) denote the optimal value for (3) and \( \overline{V} \) denote the optimal value for (4). We find that for centralized allocation, the effect of the change in the objective function is small. Service curves for fixed fractional and adaptive whole storage placement policies, \( f_{m,0} \) and \( f_{m,1} \), for a video \( m \) requested by 200 peers are shown as the two solid curves in Figure 4. The expected total download rate provided by the caches is plotted versus the number of copies of the video stored in the cache system. The maximum of the two solid curves is \( f_m \) and the dotted curve on top is its concave hull, \( \overline{f}_m \).

![Graph](image2.png)

**Fig. 3.** Single video performance of a cache system of 50 caches, 2000 peers and four random connections per peer

In these plots, the single video placement service curves of fixed whole and fixed fractional storage placement policies remain the same across videos of all popularity, because of the non-adaptive property of these two policies.

The service curves of adaptive whole and adaptive fractional storage placement policies drop down and converge to those of their corresponding fixed placement policies as videos become more popular, due to the law of large numbers. That is, the demand for more popular videos among caches is more balanced, leading to less variation between best and average video placements. Thus, the adaptive property of placement policies becomes less beneficial.

The service curve of the adaptive whole storage placement policy rises up and converges to that of the adaptive fractional storage placement policy as videos become less popular, due to the imbalance of random cache-peer connections. That is, the demand for less popular videos among caches is very unbalanced as few caches have significantly higher demand. Thus, the adaptive placement of whole copies of a video is nearly optimal because the adaptive fractional storage placement policy tends to place near integer fractions of videos.

**VI. HYBRID MULTIPLE VIDEO PLACEMENT POLICY**

The adaptive fractional storage placement policy serves at least as many peers as the other three policies. However, adaptive fractional storage suffers overhead due to the need to encode and decode to provide fractional storage and to adapt to current demand. Hence, we seek another policy serving nearly as many peers, but with less overhead. As observed in section V, for videos with low popularity, the adaptive whole storage placement policy serves nearly as many peers as the adaptive fractional storage placement policy. And for popular videos, the fixed fractional policy serves nearly as many peers as the adaptive fractional storage placement policy. This suggests using a hybrid policy that follows the adaptive whole placement policy for videos with low popularity and the fixed fractional placement policy for popular videos.

Let \( V^* \) denote the optimal value for (3) and \( \overline{V} \) denote the optimal value for (4). We find that for centralized allocation, the effect of the change in the objective function is small. Service curves for fixed fractional and adaptive whole storage placement policies, \( f_{m,0} \) and \( f_{m,1} \), for a video \( m \) requested by 200 peers are shown as the two solid curves in Figure 4. The expected total download rate provided by the caches is plotted versus the number of copies of the video stored in the cache system. The maximum of the two solid curves is \( f_m \) and the dotted curve on top is its concave hull, \( \overline{f}_m \).

There can be at most one crossover between \( f_{m,0} \) and \( f_{m,1} \) because \( f_{m,0} \) is concave and \( f_{m,1} \) is linear up to the point all peers requesting video \( m \) are served. We observe from Figure 4 that if a crossover occurs, the concave hull, \( \overline{f}_m \), is linear over an interval \([A_m, B_m]\) and the crossover point \( T_m \) falls within the interval. Since \( \overline{f}_m \) is a concave upper bound to \( f_m \), \( \overline{V} \) is an upper bound on \( V^* \) and the solution to (4)
can be obtained by applying the hybrid algorithm (a modified version of Algorithm 1 for multiple videos with a tie breaking rule), which can be found in the full version.

The resulting hybrid algorithm determines the number of copies, $\mathcal{C}_m$, of each video $m$ to store yielding the solution $\mathcal{C}$ to (4). Based on $\mathcal{C}$ and the definition of $f_m$, we can find which placement policy to use for each video $m$: $\tilde{\theta}(m) = \arg \max_{\theta \in [0,1]} \{ f_m(\mathcal{C}, \theta) \}$. Thus, $V^*$ can be approximated by $U(\mathcal{C}, \tilde{\theta})$. The extent of suboptimality of the placement can be shown to be small and does not grow with the number of caches or the number of videos. Let $m^*$ denote the video selected in the final iteration of the hybrid algorithm, before the storage constraint is met. Define $\Delta(m) \triangleq \max_C \{ f_m(C) - f_m(\mathcal{C}_m) \}$, which is the maximum difference between $f_m$ and its concave hull.

**Proposition 6.1:** The extent of suboptimality of the placement found by the hybrid algorithm is bounded as follows: $V^* - U(\mathcal{C}, \tilde{\theta}) \leq \Delta(m^*)$.

### VII. MULTIPLE VIDEO POLICY SIMULATIONS

To compare the algorithms for a large system, we simulated the algorithms for the system parameters used in [10]: 40,000 peers, 50 caches, and 2000 videos following a 0.8 Zipf distribution. Each peer is connected to 4 caches selected randomly and uniformly, forming a random graph. The system’s total storage constraint is 2.5 times the entire video catalog - 5000 copies. The total numbers of peers served by the cache system are shown in Table I for each multiple video placement policy. The adaptive fractional storage placement policy yields the maximum number of peers that are served by the cache system without help from the server for multiple videos.

<table>
<thead>
<tr>
<th>Multi-video placement policy</th>
<th>Total # of peers served</th>
<th>Fraction of optimal performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed whole</td>
<td>21747</td>
<td>69.9%</td>
</tr>
<tr>
<td>Fixed fractional</td>
<td>26746</td>
<td>84.6%</td>
</tr>
<tr>
<td>Adaptive whole</td>
<td>30092</td>
<td>95.8%</td>
</tr>
<tr>
<td>Adaptive fractional</td>
<td>31413</td>
<td>100%</td>
</tr>
<tr>
<td>Hybrid</td>
<td>31008</td>
<td>98.7%</td>
</tr>
</tbody>
</table>

For placement of multiple videos according to the hybrid storage placement policy, roughly 100 of the most requested videos are stored using the fixed fractional storage placement policy and the remaining videos are using adaptive whole storage placement policy. The total number of peers served by the cache system achieves about 99% of the optimal policy (adaptive fractional storage placement policy)’s performance.

### VIII. CONCLUSION AND FUTURE WORK

In this paper, we studied the content placement problem for cache delivery VoD systems. Instead of performing content placement analysis on the whole system with multiple videos, we approached the content placement problem by analyzing the decoupled systems with only a single video. With the insights gained from single video placement policy analysis, we returned to the original content placement problem by constructing a general method for the four placement policies for multiple videos, which places copies of videos so as to maximize the total expected download rate for all users from the caches. We provided analytical and simulation results that answer the key question of how many more peers can be served using fractional storage or adaptive placement. Finally, based on these results, we proposed a hybrid storage placement policy for multiple video placement, which is a lower complexity alternative to the optimal content placement policy serving nearly as many peers.

In practice, the hybrid algorithm can be simplified without much loss in performance by estimating in advance which videos to serve using the fixed fraction policy and which to serve using the adaptive whole placement policy.

### REFERENCES