The $S^2$ Piggybacking Policy

ROBERTO DE A. FAÇANHA  
facanha@dcc.unicamp.br

NELSON L. S. DA FONSECA  
nfonseca@dcc.unicamp.br

PEDRO J. DE REZENDE  
rezende@dcc.unicamp.br

Institute of Computing, State University of Campinas, Campinas S.P., Brazil, PO Box 6176

Abstract. In video-on-demand systems, users expect to watch a film right after its selection. Nonetheless, such a short response time is feasible only if there is available bandwidth. In line with that, several techniques have been proposed to reduce the huge bandwidth demand on video servers. In this paper, we introduce the Piggybacking policy $S^2$, which adds a second level of optimization to the snapshot policy. Moreover, we introduce a heuristic to reduce the complexity to generate the tree of superimposed video streams.

Keywords: piggybacking, video-on-demand

1. Introduction

Video on Demand (VoD) is one of the most promising applications in the future Broadband Integrated Services Digital Networks (B-ISDN). However, since it is extremely demanding on bandwidth, its availability in large scale requires the use of techniques for reducing this demand. Such techniques take into consideration the probability of a set of requests for popular videos (hot videos) getting to the system within a relatively short time interval, making it possible to provide a single video stream for all these requests.

Batching is a technique [4, 3, 9] which starts a new video stream by grouping all pending requests for a particular video in a certain time window. The main disadvantage of this technique is the delay introduced between the request and the beginning of exhibition of the video, which might lead the user to drop the request. Piggybacking is a technique [7, 8, 1] based on the fact that alterations of up to 5% in the display rate of a film are not perceptible to the user. In this way, requests can be promptly initiated and, by varying the display rate of subsequent video streams of the same film, we can superimpose the video streams as soon as they display the same frame. By doing so, we can provide only one video stream for all superimposed streams.

This paper presents the $S^2$ Piggybacking policy, which was discussed in the seminal paper [5]. The $S^2$ policy is a generalization of the snapshot Algorithm policy [1]. Furthermore, we propose an heuristic for reducing the complexity of building merging tree.

The remaining of this paper is organized as follows: section 2 presents the snapshot policy. Sections 3 and 4 introduce the $S^2$ policy and the proposed heuristic, respectively. Section 5 presents some conclusions. Appendix A shows the complexity analysis and appendix B presents the source code for the heuristic.
2. The Snapshot Algorithm

Snapshot is a Piggybacking policy [1] which attempts to minimize the number of frames shown for a given set of video streams. The computations performed by Piggybacking policies can be regarded as a binary tree (merging tree) in which the leaves correspond to streams, inner nodes are merges and the root is the final merge which forms the resulting stream of the set (Figure 1). Therefore, the number of possible trees generated by a set of streams constitutes the number of Piggybacking policies potentially optimal and is given by the $(n - 1)$st Catalan [1, 6] number, that is:

$$Catalan(n - 1) = \frac{1}{n!} \cdot \frac{(2n - 2)!}{(n - 1)!}$$

which implies that it is not viable to perform exhaustive search in an attempt to find an optimal strategy and its corresponding binary tree.

![Figure 1. Possible merging trees for four streams.](image)

The Snapshot policy builds an optimal merging tree in the following way: consider a set of $n$ streams of a single video, comprised of $L$ frames, and their positions being given by $f_1, f_2, \ldots, f_n$, where $f_1 \geq f_2 \geq \ldots \geq f_n$, at a given instant of time $T$. The minimum and maximum rates of frames per second of the video streams are denoted $S_{\text{min}}$ and $S_{\text{max}}$, respectively. Let $i$ and $j$, where $1 \leq i < j \leq n$, be two video streams. $P(i, j)$ denotes the merging position (in frames) at which streams $i$ and $j$ show the same frame. Since stream $i$ has speed $S_{\text{min}}$ and stream $j$ has speed $S_{\text{max}}$, the merging position is given by:

$$P(i, j) = f_i + \frac{S_{\text{min}} \cdot (f_i - f_j)}{S_{\text{max}} - S_{\text{min}}}$$

Let $C(i, j)$ be the cost of a Piggybacking policy and $T(i, j)$ the corresponding binary tree. The cost of a given stream is given by

$$C(i, i) = L - f_i.$$