Presentation Lag Reduction by Scheduling Media Objects for Auto-assembled Multimedia Presentations from Educational Digital Libraries

Feng-Cheng Lin[1], Chien-Yen Lai[1], Pai-Hsun Chen[2], and Jen-Shin Hong[1,*]

[1] Department of Computer Engineering, National Chi-Nan University, Taiwan
[2] Department of Computer Engineering, Nan-Kai Institute of Technology, Taiwan
jshong@ncnu.edu.tw

Abstract. This study investigates the optimization of the ordering of retrieved media objects from educational multimedia repositories for a continuously-played presentation such that the total presentation lag through a slow network is minimized. We present a computation-efficient NEH-based heuristic algorithm that can obtain a near-optimal solution with minimal computation time. A simulation experiment shows the average gaps between the idle rate of heuristic solutions and randomly generated sequences are around 26.3%. The results indicate the proposed heuristic solution can significantly reduce the presentation lag as compared to a random ordering approach that is commonly applied in conventional multimedia repositories.

Keywords: Multimedia presentation, repository, scheduling, flowshop.

1 Introduction

Over the last decade, there is a tremendous growth in education resource repositories of learning materials. Typically, a query to a specific educational learning material repository often retrieved a bag of relevant multimedia items. To present the retrieved media objects, while many repositories typically support interfaces for a user to "click and play" the interested items one by one for downloading and presentations, there are also interests that aim to dynamically compose the media items selected into a continuously-played TV-like multimedia presentation. A continuously-played presentation particularly suits well for hand-held portable devices with which the input interfaces are usually less easy to operate than a mouse. In a dynamically generated presentation, often, the multimedia servers randomly or chronologically push the content to the users. Since there is a delay between the points when the user clicks on a presentation and when it is played, the user often experiences certain idleness while waiting for the transmission of the media objects. To present an auto-assembled multimedia document online, an important concern for optimally scheduling the objects is the "total latency" experienced by the user during a presentation. The presentation latency is particular important when the multimedia

* Corresponding author.

D.H.-L. Goh et al. (Eds.): ICADL 2007, LNCS 4822, pp. 454–457, 2007.

presentation is delivered through a low-bandwidth communication channel, which might be caused by bandwidth limitation in the backbone network, a busy server, or a client with a slow "last-mile" connection. To cope with such a delay issue, a commonly used strategy is by the "streaming" technology. However, in a slow network environment where the media data consumption rate is higher than the data transmission rate, there are still intermittent idle durations during a streaming-based presentation. Besides, while the streaming technology has been widely and successfully applied in commercial web sites, its applicability in real-life application in campuses is somehow more restricted. In reality, many teachers do not have knowledge and tools to author and distribute streaming-based multimedia objects.

The media objects in a typical educational repository cover a wide spectrum of modalities with different formats including text, images, audio, video, and vector graphics. For a same amount of data transmitted, the anticipated presentation time for different media types actually differs drastically. How to order them in a prefetch-enabled continuously-played presentation to reduce the total presentation latency in a bandwidth-limited environment is interesting to investigate. This paper explores techniques for optimizing the ordering of media items to provide better Quality of Service (QoS). The sequence optimization techniques can be integrated into a server for dynamically assigning the order of the selected media items to be delivered. In the following, the problem will be formulated and solved using techniques adapted from conventional studies in operational researches.

2 A Heuristic Solution for Approximating the Optimal Sequence

In principle, the media scheduling problem addressed here can be mapped to a "two-machine flowshop problem" [1-2] that aims to optimize a scheduling of sequential jobs to be processed on two machines. The objective is to minimize the completion time of all jobs. Table 1 gives a mapping of the parameters between the media scheduling problem and the two-machine flowshop problem. In this study, we present a computation efficient heuristic algorithm that can be used to generate feasible near-optimal approximate solutions. The performance of the heuristic approach will be evaluated empirically using simulations. In the following, we describe the heuristic approach for solving this sequence optimization problem with a limited buffer size.

Table 1. Correspondence between media sequence problem and two-machine flowshop problem

Sequence optimization problem	Two-machine flowshop problem	Notation
A set of n media items	A set of n jobs	M
A media item	A job	m_i
Server	First machine	*server*
Client	Second machine	*client*
Transmission time for a media item	Processing time on the first machine	a_i
Playback time for a media item	Processing time on the second machine	b_i
Completion time for a media item	Completion time for a job	c_i
Presentation span	Makespan (maximum c_i of M)	$C_{\max}$

2.1 NEH-Based Approach for Real Time Optimization

A recent study [3-4] compared 25 well-known heuristic approaches for obtaining near-optimal solutions for flowshop problems and showed that the NEH approach frequently outperforms other approaches on several benchmark problem sets. Furthermore, the NEH approach is rather straight forward and easy to implement. Therefore, we have applied the NEH heuristic solution for the media sequence optimization problem. We outline the major steps of the NEH heuristic as follows.

Step 1: For each m_i, compute $c_i=a_i+b_i$. Sort the media jobs by non-increasing c_i.
Step 2: Take the first two media items in the sorted M, and order the two items such that the partial completion time of the first two items is minimized.
Step 3: Insert the 3 rd jobs into the partial schedule in the previous step. Since there are two existing jobs in step 2, there are three possible positions for the insertion of the 3rd job. Among these 3 positions, select the position which minimizes the partial completion time under the given buffer size.
Step 4: For k=4 to n do
Insert the k-th job into the previous partial schedule. There are k possible positions for this insertion. Select the position which minimizes the partial completion time under the given buffer size.
Step 5: Output the final sequence from step 4.

2.2 Performance Evaluations for the NEH Heuristic Solutions

This section presents the computational experiments designed to evaluate the effectiveness and efficiency of the proposed NEH-based heuristic approximations for searching near-optimal media sequences. For each job set, the processing times a_i and b_i were randomly generated numbers ranging between 1 and 100. The transmission bit rate is 160KB/s. The simulation codes were written in C++ language, and the experiments were performed on an IBM xSeries x206m computer. We conducted a series of computational experiments with different problem sizes, from 15 to 50 media items. The problem generation procedure yields the test problems that possibly encompass a wide variety of real life scenarios of online multimedia applications. For each experiment, 50 different media sets are used. Three different buffer size constraints, 16,000 KB, 22,400KB, and 30,720 KB respectively, were given for the experiments. A typical computation time using the NEH algorithm for problems with 20 media objects is less than 0.0005 seconds of CPU time. For each media set, the idle rate is defined as the ratio of the total idle time and the total transmission time of the media items, that is,

$$Idle\_rate = \frac{C_{\max} - \sum b_i}{\sum a_i} * 100\% \quad (1)$$

An *Idle_rate* close to 1 refers to a case where the media items are badly ordered such that the playbacks are mostly halted during the downloading time. Table 2 lists the *Idle_rate*s of the NEH heuristic solutions and randomly generated sequences in problems with different number of media items. The average *Idle_rate*s of the NEH

solutions are 45.2%, 14.4%, and 6.3% for problems with 16,000KB, 22,400KB, and 30,720KB buffer constraints respectively. The average *Idle_rate*s of the random (RAN) are 75.0%, 47.6%, and 23.0% respectively. The average gaps between the *Idle_rates* of NEH and RAN are around 26.3%. These results indicate the NEH solutions significantly reduce the presentation lags as compared to random sequences.

Table 2. Computation results on the *Idle_rate* (%)

n	Buffer	16,000KB	22,400KB	30,720KB	*n*	Buffer	16,000KB	22,400KB	30,720KB
15	NEH	49.9	18.8	9.2	35	NEH	42.5	13.0	5.9
	RAN	76.1	49.7	26.1		RAN	73.7	46.5	22.5
20	NEH	46.3	15.5	6.4	40	NEH	43.0	11.9	5.2
	RAN	74.9	46.3	25.3		RAN	74.8	44.9	22.3
25	NEH	45.9	16.4	7.5	45	NEH	46.1	12.9	4.9
	RAN	74.8	46.6	23.8		RAN	76.3	46.8	21.0
30	NEH	46.6	15.7	8.0	50	NEH	40.9	10.8	3.6
	RAN	76.0	48.1	23.4		RAN	73.4	44.1	19.6

3 Conclusions

This study investigated the optimization of the sequences of retrieved media objects such that the total presentation lag of a continuously-played multimedia presentation through a slow network is possibly minimized. Aiming for the real time online applications, we present a computation-efficient NEH-based heuristic algorithm that can obtain a near-optimal solution with minimal computation time. Overall speaking, the average gap between the idle rates of heuristic solutions and randomly generated sequences is around 26.3%. These simulation results indicate that the NEH-based heuristic solutions can significantly reduce the presentation lags as compared to a random ordering approach which is commonly applied in conventional multimedia applications.

Acknowledgments. This work is supported by the National Science Council of Taiwan, under grant NSC-94-2422-H-260-002.

References

1. Pinedo, M.: Scheduling Theory, Algorithm, and Systems, 2nd edn. Prentice Hall, New Jersey (2002)
2. Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, R., A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics 5(1), 287–326 (1979)
3. Nawaz, M., Enscore, E., Ham, I.: A heuristic algorithm for the m-machine, n-job flow shop sequencing problem. OMEGA, International Journal of Management Science 11, 91–95 (1983)
4. Ruiz, R., Maroto, C.: A comprehensive review and evaluation of permutation flowshop heuristic. European Journal of Operational Research 165, 479–494 (2005)