Efficiently Scheduling Tape-resident Jobs*

Jing Shi, Chunxiao Xing, Lizhu Zhou
Department of Computer Science and Technology
Tsinghua University
Beijing 100084, P.R.China
Shijing@mails.tsinghua.edu.cn, {xingcx, dcszlz}@tsinghua.edu.cn
Tel: +86-10-62789150

Abstract
Many large-scale data-intensive applications need to use tape library to manage large data sets, thus it is critical to study the online access techniques of tape library. The focus of this paper is on efficient tape-resident jobs scheduling, which is the key technique for improving performance of tape storage systems. We present several scheduling algorithms for tape-resident jobs, discuss the effectiveness of scheduling policies under cache-limited and cache-unlimited condition, and show the results of simulation experiments.

1 Introduction

Many data repositories are expected to become huge, possibly counted by terabytes in size. Examples of such repositories include terabyte-level Telecommunications Call Detail Warehouse, petabyte-level Digital Libraries, exabyte-level National Medical Insurance Records, Zettabyte-level Spatial and Terrestrial Database and video and Audio Data Archives[1][2]. The management of such large data sets requires the use of tertiary storage, typically implemented by using tape libraries. As a result, accessing, analyzing, mining, and other data-intensive applications can comprise of many tape-resident jobs that retrieve either wholly, or in part, data from tapes.

Tape library is characterized by (1) the use of removable tape media and a robot arm, (2) sequential access of data, and (3) the performance bottleneck caused by tape access. Tape-resident job usually consists of more than one request, each of which must be completed before the job is finished, and uses disk cache space to store the data of its completed requests. To improve the performance of tape-resident jobs, we have to consider the following two problems -- the accessing latency of tape library, and the capacity limitation of disk cache for storing the retrieved data from tapes.

Previous studies mostly focus on the request scheduling of tape library to improve performance of robotic storage library[3][4][5][6]. But our goal is to schedule the jobs consisting of a set of requests to minimize the completion time of the whole job. A study closely related to ours is the one in which the scheduling problem of tape-resident jobs is reduced to well-known flow-shop scheduling[7]. However, it doesn't consider the optimal scheduling of tape libraries.

In this paper, we will introduce better scheduling strategies for executing tape-resident jobs. We will discuss how to improve the performance of tape-resident jobs by optimized

* This research is sponsored by the National Grand Fundamental Research 973 Program of China under Grant No.G1999032704

I/O performance of tape library, and discuss the effectiveness of scheduling policies under cache-limited condition or cache-unlimited condition by simulation study. Section 2 gives the scheduling problem description of tape-resident jobs. The scheduling algorithms will be presented in Section 3 and the simulation results for performance comparison of scheduling algorithms will be given in Section 4. Finally, Section 5 concludes the paper.

2 Problem Description

A tape-resident job consists of a set of requests, each of which is a read operation for a set of continuous blocks on a tape. We assume that the requests are independent of one another, that is, requests don't need to be executed in some forced order. The reason is that the access of tape library is much slower than that of disk, if processor begins to execute the job before the data involved in by its requests are all loaded into disk cache, then the job is possibly blocked for waiting unloaded requests. So we reduce the execution principle of tape-resident jobs to a simple form, that is, the job doesn't begin to execute until the data of its requests are all loaded into disk. This assumption means that the data of requests may be loaded by any order. The following Fig.1 is the description model of tape-resident jobs.

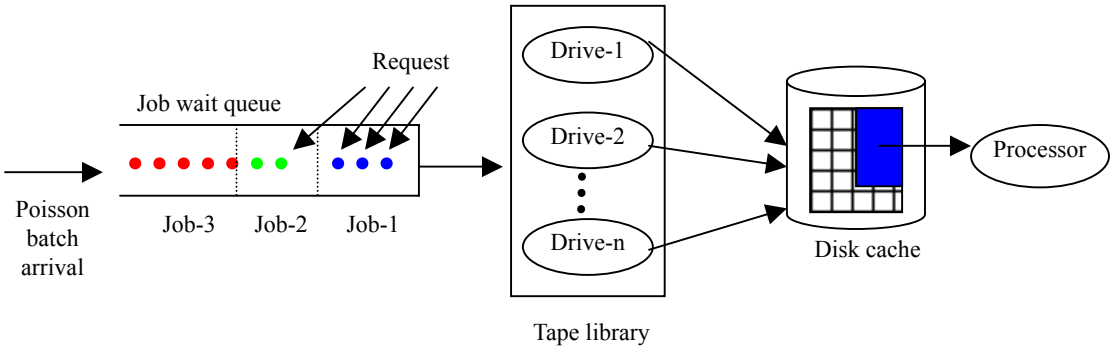

Fig. 1 The description model of tape-resident jobs

Since a job of several requests may involve more than one tapes, combining jobs that access the same media will make system process as much requests as possible in a tape schedule. One problem is that if jobs are not properly scheduled, the disk cache may be run out quickly. Therefore, it is critical to study the correlation between tape drive utilization and disk capacity limitation for tape-resident job scheduling. To do so, we consider the following optimization policies when designing tape-resident job scheduling algorithms:

- To improve the I/O performance of tape library
- To reduce resident time of data of jobs on disk cache
- To coordinate the input and output throughput of jobs to or from disk cache

3 Scheduling Algorithmic Issues

We study our scheduling problems under two kinds of restrictive conditions respectively: cache-limited and cache-unlimited. The former means the selection of scheduling policies must take the available space on disk cache into consideration, and the later assumes that

there is enough space of disk cache for scheduling. We first present five scheduling algorithms under the second condition, and then discuss these algorithms with the first condition of constraint. The algorithms focus on two key points: **tape selection policy**, and **scheduling list creation** (a scheduling list is an ordered list of requests for a selected tape).

(1) FCFS (First Come First Service). This algorithm services the jobs in the order of arrival, and always chooses the tape that the first request in job wait queue accesses to for next execution. The scheduling list of selected tape includes all requests that belong to the job and access the selected tape. These requests will be executed within one sweep of the tape.

(2) Max-EBW (Maximum Effective BandWidth). This policy improves the scheduling of tape-resident job in maximizing I/O performance of tape library. It always chooses the tape with maximum effective bandwidth for the next execution. The effective bandwidth of a tape is defined to be the total number of bytes transferred from the tape divided by the number of seconds consumed to perform this tape schedule.

(3) FCFS-PICKUP. This algorithm uses simplest tape selection policy--FCFS, namely, it always selects the tape to be accessed by the first request of a job in the wait queue, and then the algorithm inserts all requests of other jobs in the wait queue that will access the selected tape into its scheduling list, which is called the *PICKUP* policy for scheduling list creation.

(4) DYN-PICKUP. This algorithm has similar tape selection and scheduling list creation as FCFS-PICKUP. Besides this, it particularly considers the new arrival jobs. When the requests belonging to a new arrival jobs are trying to access the blocks on online tape that the tape head will pass over during the current sweep, they will be inserted into the running scheduling list. This is the dynamic policy for scheduling list creation.

(5) TUNING-PICKUP. This algorithm makes FCFS-PICKUP scheduling tunable. It uses *PICKUP intension factor F*, which indicates that PICKUP scheduling is only applied among the first *F* waiting jobs in the job wait queue, to tune the scale of scheduling list. Obviously, larger *F* means both larger cache occupation, and quicker response time. The selection of proper *F* value is the difficult point of this algorithm. Currently, we determine the *F* value by simulation experiments. A proper method for F value selection will be studied.

Above algorithms have different cache requirement: FCFS needs least cache space; TUNING-PICKUP may tune the size of cache occupation by changing PICKUP intension factor F; and other algorithms use more cache space than FCFS, but are not able to tune cache requirement. The comparison details of above algorithms will be given in next section.

4 Simulation Study

In this section, we give two groups of simulation results, each of which consists of two

figures: average response time of jobs and maximum cache requirement of jobs. The simulation parameters of tape library are based on Exabyte 220 tape library with two Eliant 820 drives and twenty EXABTYE 8mm tapes. In addition, we assume that the job arrival is stochastic and follows Poisson distribution. Each job averagely consists of 8 requests that have the average size of 64M bytes. We also assume that the disk cache should at least meet the maximum storage requirement of any job. The jobs are independent of one another.

Fig.1a and Fig.1b show response time and cache occupation curves for all algorithms except for TUNING-PICKUP. From the graphs we can observe that FCFS has least cache occupation but longest response time, and other algorithms significantly improve the average response time of tape-resident jobs by optimizing I/O performance of tape library. This performance improvement from tape library optimization has an associated cost in terms of storage space. The Figure also indicates that FCFS-PICKUP is the best scheduling policy. The reason is that it uses FCFS policy to speed up job output from disk cache while it takes advantage of PICKUP policy to improve I/O performance of tape library. Although the time performance of DYN-PICKUP policy is slightly better than that of FCFS-PICKUP, but its cache occupation is much higher than FCFS-PICKUP and Max-EBW. Its heavier workload creates proportionally larger storage requirement.

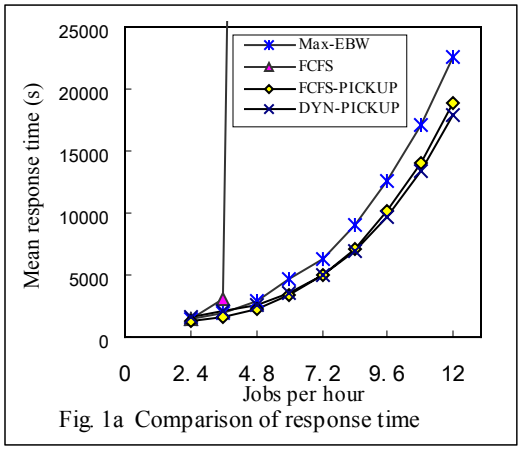

Fig. 1a Comparison of response time

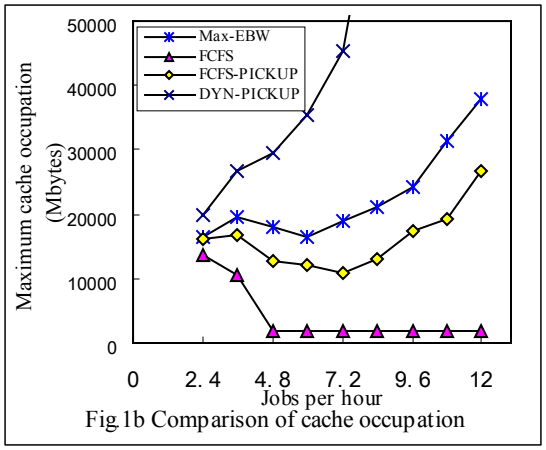

Fig.1b Comparison of cache occupation

The next simulation experiment explores the correlation between response time and cache space for FCFS-PICKUP algorithm and TUNING-PICKUP algorithm. We use *PICKUP intension factor F* to tune the size of cache occupation. This is very helpful in achieving a reasonable response time for tape-resident jobs when cache space is limited. Fig2a and Fig.2b illustrate when properly tuned, the time performance of TUNING-PICKUP is close to that of Max-EBW, but its space occupation is significantly reduced.

5 Conclusions

This paper discusses some efficient scheduling algorithms for tape-resident jobs. Our contributions include: (1) incorporate optimal I/O scheduling policies of tape library into the scheduling of tape-resident jobs so as to improve performance of tape-resident jobs by increasing the data throughput of tape library processing; (2) design better algorithm

FCFS-PICKUP for cache-unlimited system and TUNING-PICKUP for cache-limited system. The future work is to give a practical evaluation method for *PICKUP intension factor F* so that we may simply select *factor F* value for TUNING_PICKUP algorithm according to both workload and cache size.

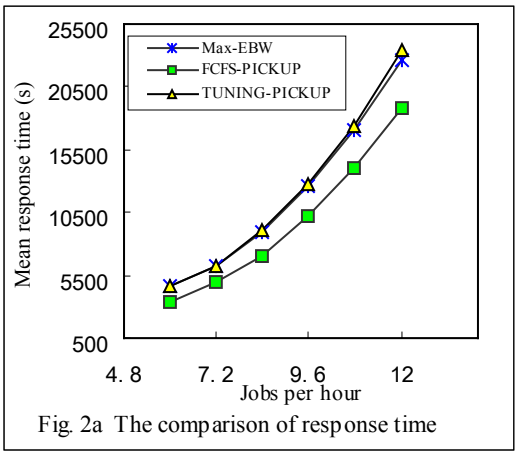

Fig. 2a The comparison of response time

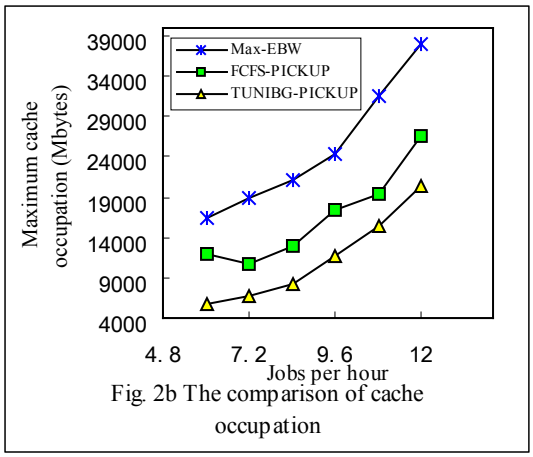

Fig. 2b The comparison of cache occupation

Reference

[1] Cariño F., Kaufmann A. and Kostamaa P., Are you ready for Yottabytes?, In Proc. of 17th IEEE symp. on Mass Storage Systems in Cooperation with the 8th NASA GSFC conf. on Mass Storage Systems and Technologies, pp. 476-485, Match 2000

[2] John Jensen, John Kinsfather and Parmesh Dwivedi. Data Volume Proliferation in the 21st Century--The Challenges Faced by the NOAA National Data Centers (NNDC), In Proc. of 17th IEEE symp. on Mass Storage Systems in Cooperation with the 8th NASA GSFC conf. on Mass Storage Systems and Technologies, pp. 335-350, Match 2000

[3] Bruce K.Hillyer and Avi Silberschatz， Random I/O Scheduling in Online Tertiary Storage Systems，In Proc. of the 1996 ACM SIGMOD Inter. Conf. on Management of Data, pp195-204, Canada, Jun 3-6 1996

[4] Bruce K. Hillyer, Rajeev Rastogi and Avi Silberschatz, Scheduling and Data Replication to Improve Tape Jukebox Performance, ICDE'99, pp. 532-541, 1999

[5] Toshihiro NEMOTO and Masaru KITSUEGAWA，Scalable Tape Archiver for Satellite Image Database and its Performance Analysis with Access Logs—Hot Declustering and Hot Replication--，In Proc. of 16th IEEE symp. on Mass Storage Systems in Cooperation with the 7th NASA GSFC conf. on Mass Storage Systems and Technologies, pp. 59-71, 1999

[6] Shi Jing and Zhou Lizhu, Dynamic Scheduling and Tuning to Improve Online Tape Library Performance, In Proceedings of the 6th International Conference for Younger Computer Scientists (ICYCS'2001), pages120-124, Oct. 2001

[7] Sachin More, S. Muthukrishnan and Elizabeth Shriver, Efficiently Sequencing Tape-resident Jobs, In Eighteenth ACM Symposium on Principles of Database Systems, 1999