Efficiently Scheduling Tape-resident Jobs

Jing Shi, Chunxiao Xing, Lizhu Zhou

Shijing@mails.tsinghua.edu.cn,

{xingcx, dcszlz}@tsinghua.edu.cn

Department of Computer Science and Technology

Tsinghua University

Background

- Many data repositories are expected to become huge counted by terabytes in size
- The management of such large data sets requires the use of tertiary storage, typically by using tape libraries
- Many data-intensive applications can comprise of many tape-resident jobs that retrieve either wholly, or in part, data from tapes

What is the tape-resident job?

- 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
- The requests are independent of one another, that is, requests don't need to be executed in some forced order
- The job is consider complete only when its last request has loaded its data into disk cache

Problem Description(1)

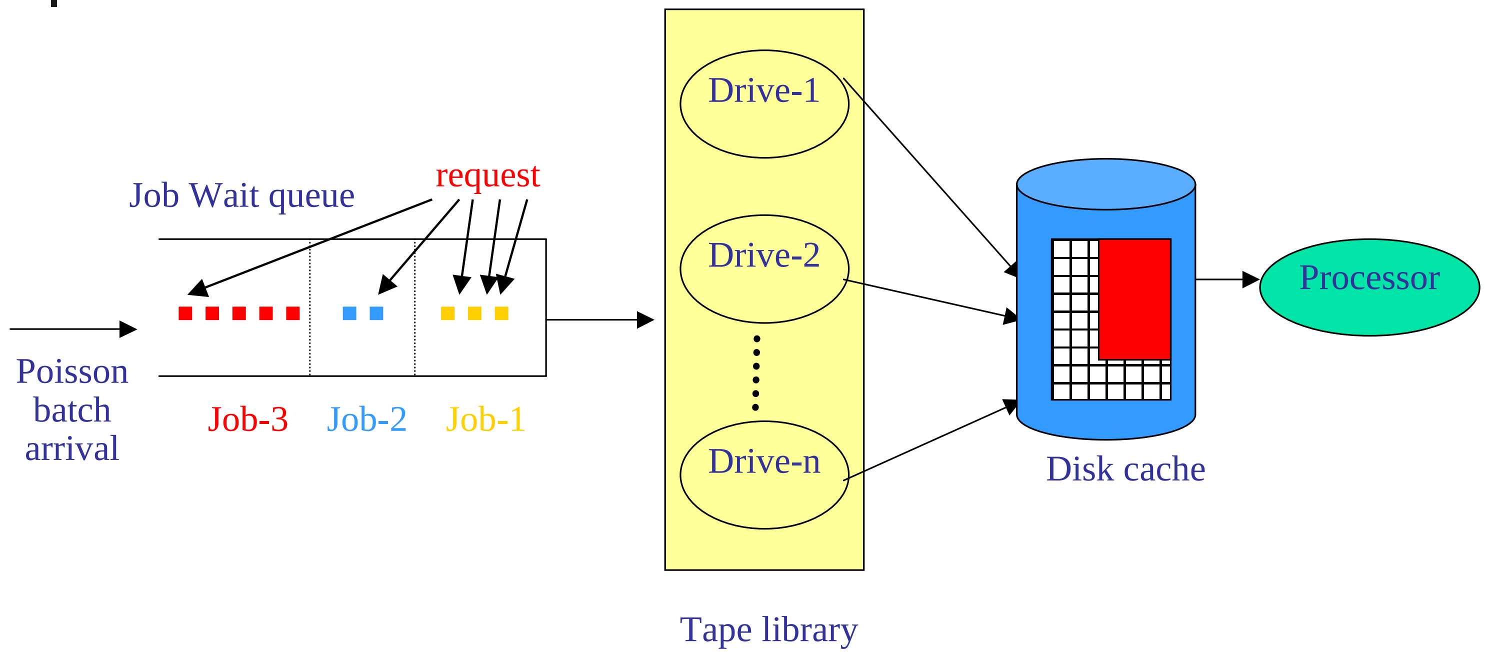

Fig. 1 The description model of tape-resident jobs

Problem Description(2)

- Jobs are Poisson batch arrival
- All loaded data of a job are stored on disk cache before all requests of the job are finished
- The job will not be executed by processor until the data of its requests are all loaded into disk cache

Optimization Goals

- To improve the I/O performance of tape library
  - To combine jobs that access the same media into a tape scheduling
- To reduce resident time of data of jobs on disk cache
  - First Come First Service
- To coordinate the input and output throughput of jobs to or from disk cache
  - To study the correlation between tape drive utilization and disk capacity limitation

Scheduling Algorithmic Issues(1)

- FCFS (First Come First Service)
  - To service the jobs in the order of arrival, and always choose the tape that the first request in wait queue accesses to for next execution
- Max-EBW (Maximum Effective BandWidth)
  - To choose 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

Scheduling Algorithmic Issues(2)

- FCFS-PICKUP
  - To select the tape to be accessed by the first request of a job in the wait queue
  - To insert all requests of other jobs in the wait queue that will access the selected tape into its scheduling list
- DYN-PICKUP
  - 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

Scheduling Algorithmic Issues(3)

- TUNING-PICKUP
  - To make FCFS-PICKUP scheduling tunable
  - To use *PICKUP intension factor F* to tune the scale of scheduling list.
  - *PICKUP intension factor F* indicates that PICKUP scheduling is only applied among the first *F* waiting jobs in the job wait queue,
  - Larger *F* means both larger cache occupation, and quicker response time

Simulation Studies(1)

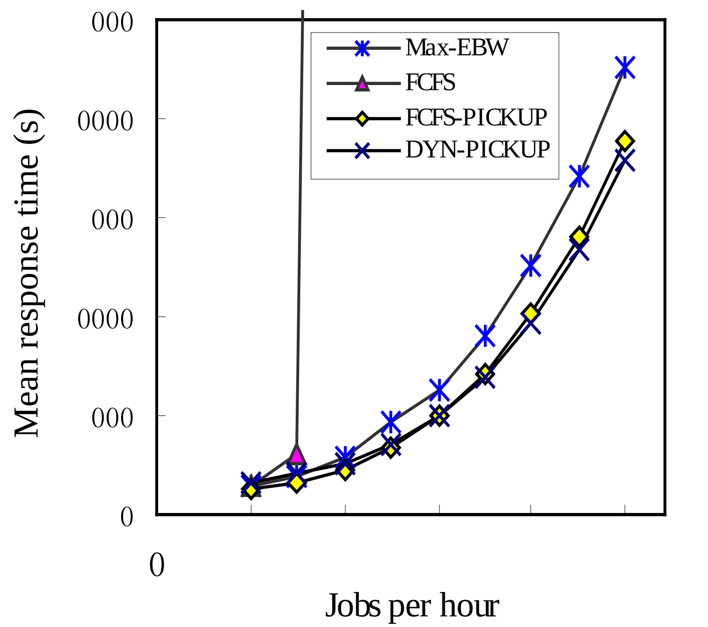

Fig. 1a Comparison of response time

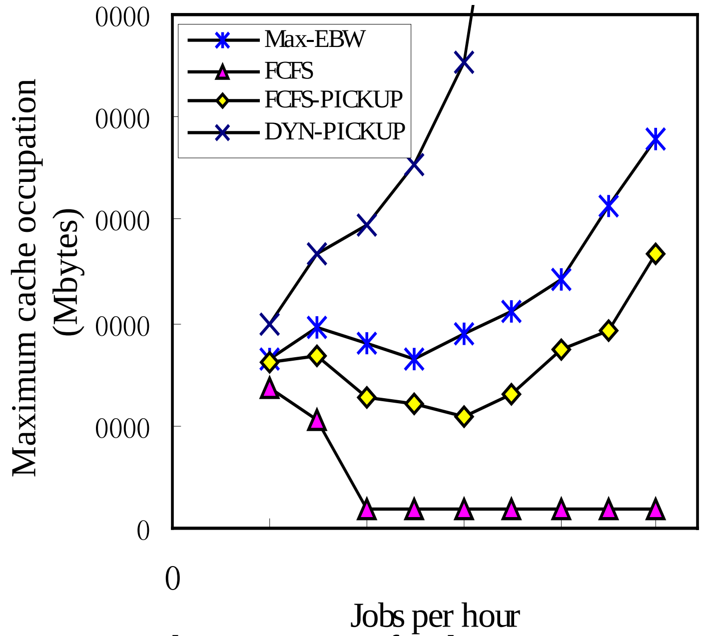

Fig.1b Comparison of cache occupation

Simulation Studies(2)

- 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
- The time performance of DYN-PICKUP is best, but its cache occupation is much higher than both FCFS-PICKUP and Max-EBW
- FCFS-PICKUP is the best scheduling policy from tradeoff viewpoint with time performance and cache occupation

Simulation Studies(3)

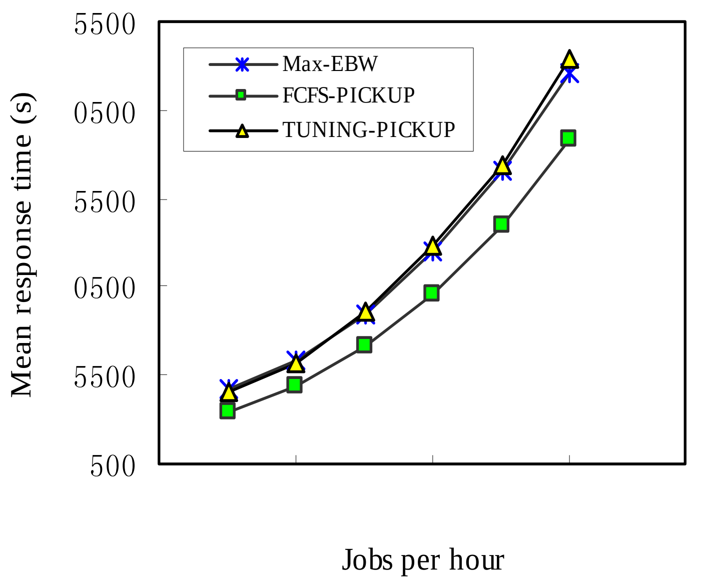

Fig. 5a The comparison of response time

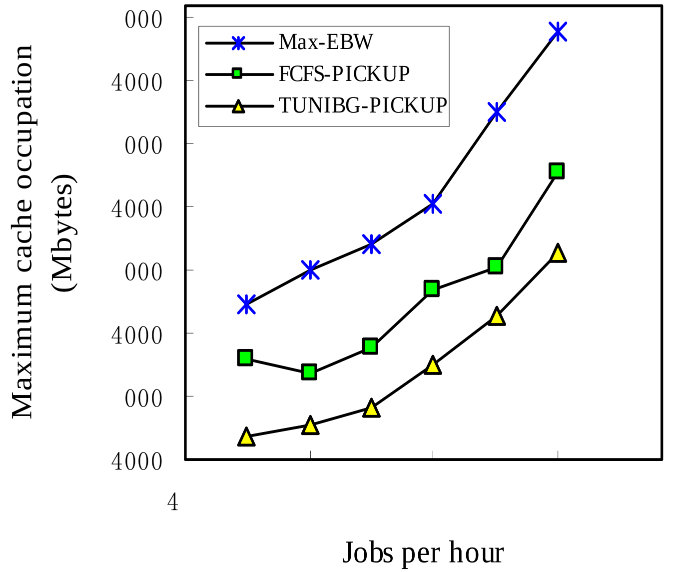

Fig. 5b The comparation of cache occupation

Simulation Studies(4)

- TUNING-PICKUP algorithm uses *PICKUP intension factor F* to tune the size of cache occupation when cache space is limited

- When properly tuned, the time performance of TUNING-PICKUP is close to that of Max-EBW, its space occupation is significantly reduced

Conclusions(1)

- Work
  - To discuss some efficient scheduling algorithms for tape-resident jobs
- Contributions:
  - To 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
  - To design better algorithm FCFS-PICKUP for cache-unlimited system and TUNING-PICKUP for cache-limited system

Conclusions(2)

- Future work
  - 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

Thank You