Promote-IT: An Efficient Real-Time Tertiary-Storage Scheduler

Maria Eva Lijding
lijding@cs.utwente.nl
+31.53.4893770

NASA/IEEE MSST 2004
12th NASA Goddard/21st IEEE Conference on Mass Storage Systems & Technologies
The Inn and Conference Center
University of Maryland University College
Adelphi MD USA
April 13-16, 2004
Structure of the Talk

- Hierarchical Multimedia Archive
- Jukebox Scheduler (Promote-IT)
- Performance Evaluation
- Conclusions
System Overview

Network

Personal MTV Server  Multimedia Database Server  Video-on-Demand Server  Backup Server  File Server (NFS)

Hierarchical Multimedia Archive

Network servers
Jukebox Architecture

- Storage capacity
- Cost/GB
- Reliability

- Switching times
- Few drives
- Shared robots
- Potential resource-constraint problems

15-04-2004
Request

Request = \{\text{deadline, asap, maxConf,} \{\text{Request Unit}\}^*\}

Request Unit = \{\text{medium, offset, size, } \Delta \text{deadline, bandwidth}\}
Scheduler Goals

- Guarantee real-time access to data
  - No hiccups or interruptions
  - Data available according to request
  - (Best-effort if access differs from request)
- Minimize response time ASAP requests
- On-line scheduling
- Minimize confirmation time
- Make good use of jukebox resources
Solution Guidelines

- Secondary storage as buffer and cache
- Detailed and flexible hardware model
- Formal model of scheduling problem
- Separate **Schedule Building** and **Dispatching**
  - Early Dispatcher
Hardware Model

- Removable Storage Media (RSM)
  - Any number
  - Different types (optical disks, tape)
  - Different types in jukebox (DVD, CD, DVD-RAM)

- Drives
  - Any number
  - Non-identical
  - Parameters depending on RSM

- Robots
  - Any number
  - Scope: Shared/dedicated/serving a set of drives and RSM
  - Functionality: Loader/Unloader/Loader-Unloader
  - Parameters depending on drives and shelves
Schedulers

- Optimal Scheduler
- Optimal Model
- Fixed Switching Model
- Dedicated Robots Model
- Minimum Switching Model
- Imperative Switching Model
- Periodic Quantum Model
- Promote-IT
- Extended Aggressive Strategy
- Extended Switch-Read Model
- Extended Conservative Strategy
- Fully Staged Before Starting
- Jukebox Early Quantum Scheduler

- 1 job per RSM + simplified hardware model
- 1 job per request + Simplified resource usage, hardware model, and request model
- Fixed clustering of RUs
- Simplified resource usage, and hardware model
Minimum Switching Model

- Flexible Flow Shop with 3 stages
  - Load, Read, Unload
  - Uncoupled Load and Unload
- Read all data from RSM before unloading
  - Minimum number of switches
- Resource constraints to guarantee mutual exclusion
Promote-IT
Characteristics

- Aperiodic scheduler
- Pipeline
- Early dispatching
- Uncoupled load and unload
- Heuristic (polynomial) scheduler
  - Branch-and-bound algorithm
  - Best-drive heuristic
Promote-IT
Scheduling Strategies

Front-to-Back

Earliest Starting Time First (ESTF)

Earliest Deadline First (EDF)

Back-to-Front

Latest Starting Time Last (LSTL)

Latest Deadline Last (LDL)

ordered by LST

ordered by deadline
Aperiodic Scheduling

![Graph showing time (sec) vs. system load (requests/hour) with different algorithms:
- JEQS
- JEQS only EQTs
- Promote-IT
- FSBS]
Pipelining

![Graph showing time (seconds) vs. system load (requests/hour) for different methods: FSBS, Promote-IT (LDL), and Promote-IT (ESTF).]
Early Dispatching

![Graph showing time (seconds) vs. system load (requests/hour)]

- Conservative
- Promote-IT (LDL)
- Promote-IT (ESTF)
Decoupled Load/Unload

![Graph showing time (seconds) vs. system load (requests/hour) for different strategies: Aggressive, Promote-IT (LDL), and Promote-IT (ESTF).]
Heuristic

![Graph showing Time (sec) vs. System load (requests/hour). The graph compares different strategies: Optimal, Aggressive, Promote-IT (LDL), and Promote-IT (ESTF).]
Heuristic (Computing Time)

- Optimal
- Aggressive
- Promote-IT (LDL)
- Promote-IT (ESTF)
Conclusions

- Flexible way to access tertiary storage
- Real-time guarantees
- Promote-IT
  - Efficient polynomial scheduler
  - Better performance than other schedulers
  - Response time near optimal
  - Can handle any type of requests and jukebox hardware