
Distributed Video Systems
Chapter 5
Issues in Video Storage and Retrieval
Part 4 - Implementation Issues

Jack Yiu-bun Lee
Department of Information Engineering
The Chinese University of Hong Kong

Contents

Jack Y.B. Lee

- 5.1 OS Real-time Support
- 5.2 Process/Threads Scheduling Issues
- 5.3 High-Performance Disk Reads
- 5.4 Case Study

5.1 OS Real-time Support

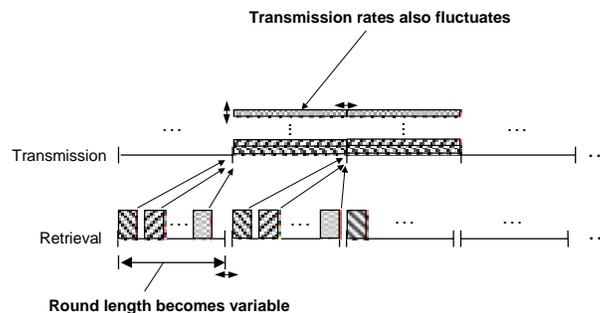
Jack Y.B. Lee

- Real-time Support for Scheduling
 - ♦ Accurate determination of time
 - E.g. for measuring the length of a service round.
 - ♦ Suspension of process/thread for an accurate amount of time
 - E.g. using sleep() to wait until the service round ends.
- Problem
 - ♦ Conventional operating systems have no real-time support.
 - BSD UNIX, System V UNIX
 - DOS, Windows, Windows95, WindowsNT, etc.
 - MacOS
 - etc, etc.

5.1 OS Real-time Support

Jack Y.B. Lee

- Example: Windows NT
 - ♦ Time determination can be very accurate (in clock cycles under Pentium CPUs).
 - ♦ But programmatic delay is very coarse (>5ms).



5.1 OS Real-time Support

Jack Y.B. Lee

- Implications
 - ◆ Implementing server-push designs become much more difficult in such operating systems.
 - ◆ More engineering margins must be used to accommodate variations in time determination and programmatic delay.
 - ◆ Service round and delay, etc. cannot be too small to avoid large errors.
- Alternatives
 - ◆ Make use of third-party hard-real-time extension for the OS, if available.
 - ◆ Use client-pull service model.
 - ◆ Use a real-time OS (e.g. QNX).

5.2 Process/Threads Scheduling Issues

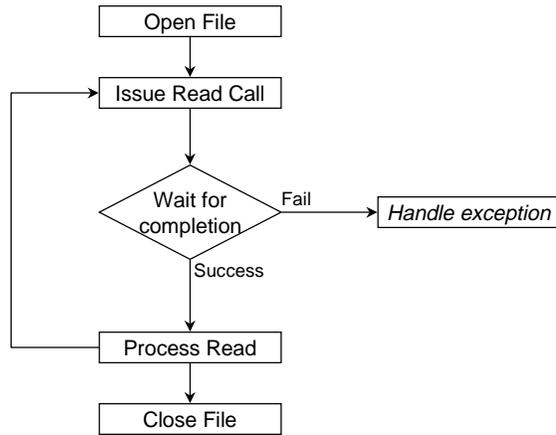
Jack Y.B. Lee

- Multitasking/Multithreaded OS
 - ◆ Processes other than the video server can compete for CPU time, disk time, as well as network time!
 - System maintenance processes like defragmentation, virus scanning, etc.
 - System processes like pagefile management.
 - ◆ Counter Measures
 - Don't put a page file in the disks that store videos.
 - Use separate disks for the operating system and videos.
 - Disable all unnecessary system functions and services.
 - Run the video server at higher process/thread priority.
 - Make use of real-time OS extension or a real-time OS instead.

5.3 High-Performance Disk Reads

Jack Y.B. Lee

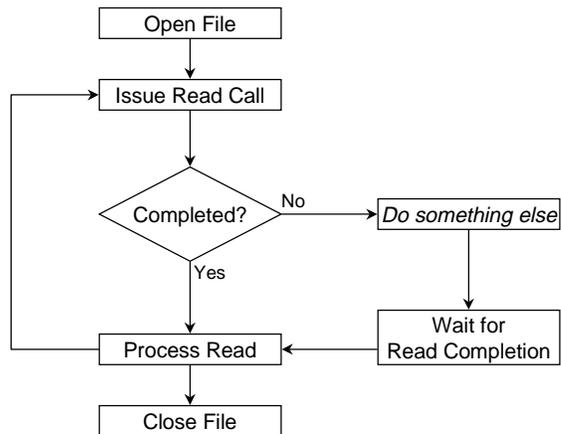
- Conventional Synchronous I/O
 - ♦ Program Flow



5.3 High-Performance Disk Reads

Jack Y.B. Lee

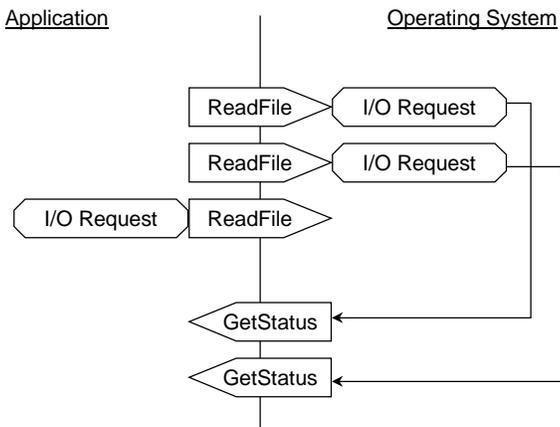
- Asynchronous I/O
 - ♦ Program Flow



5.3 High-Performance Disk Reads

Jack Y.B. Lee

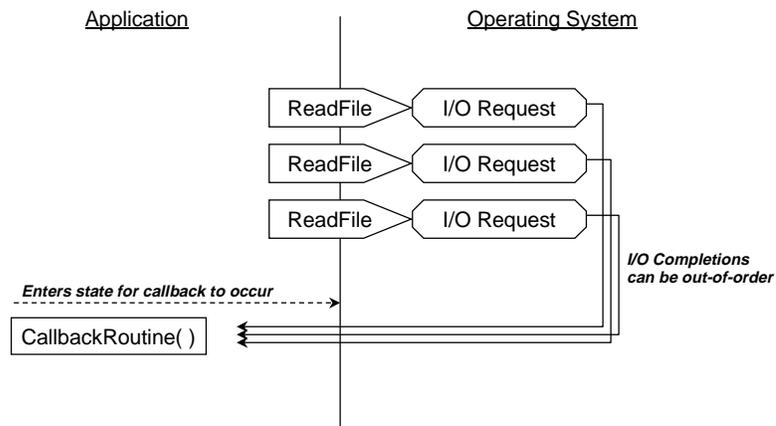
- Asynchronous I/O
 - ◆ Issuing Multiple Pending Requests
 - Application-initiated synchronization:



5.3 High-Performance Disk Reads

Jack Y.B. Lee

- Asynchronous I/O
 - ◆ Issuing Multiple Pending Requests
 - Callback synchronization:



5.3 High-Performance Disk Reads

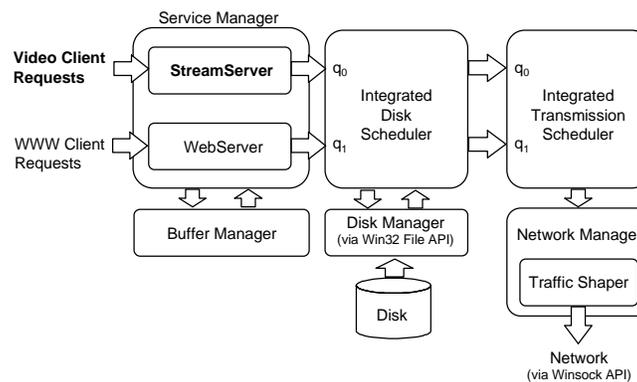
Jack Y.B. Lee

- Other Issues
 - ♦ Disk read look-ahead by the OS.
 - Look-ahead is useless and consumes extra buffers.
 - ♦ Disk caching by the OS.
 - Cached video data are unlikely to be reused and waste memory.
 - Disk caching for video retrievals should be disabled.
 - ♦ Buffer management
 - The memory allocation and deallocation functions provided by the OS or compiler usually incur more overhead than custom functions.

5.4 Case Study

Jack Y.B. Lee

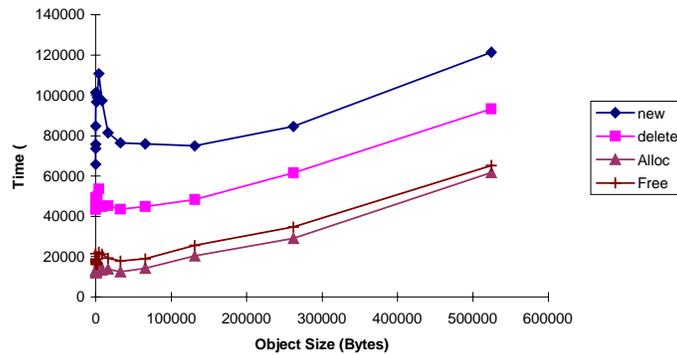
- A Multimedia Server
 - ♦ Architecture



5.4 Case Study

Jack Y.B. Lee

- A Multimedia Server
 - ◆ Performance of Buffer Manager



5.4 Case Study

Jack Y.B. Lee

- A Multimedia Server
 - ◆ Performance of Storage Server

