More on File System

CS 256/456

Dept. of Computer Science, University of Rochester

File System Prefetching

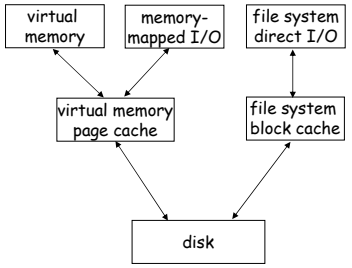
- File content is read ahead of time for anticipated use in the near future
- Often sequential (based on past access history on the file)
- What is the advantage of file prefetching?
- What is the danger of file prefetching?
- A balanced scheme that provides a competitive performance to the optimal scheme [Li et al. EuroSys 2007]

Informed Prefetching

- Informed prefetching – prefetching while utilizing some information about application data access pattern
- Application I/O hints [Cao et al. 1994] [Patterson et al. 1995]
- Automatic I/O hints based on speculative execution [Chang&Gibson 2000], [Fraser&Chang 2003]

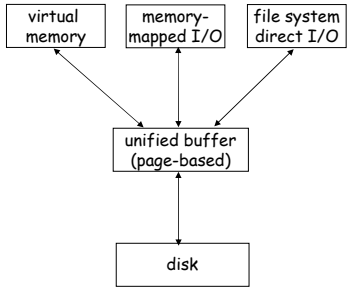
Buffer Cache in Main Memory

- Memory-mapped I/O naturally share page cache with the virtual memory system
- Problems:
  - double buffering
  - inconsistencies



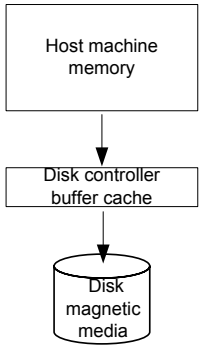
Unified Buffer Cache & Unified Virtual Memory

- A unified buffer cache uses the same page cache to store [Pai et al. 1999]
  - virtual memory pages
  - memory-mapped pages
  - file system direct I/O data



Multi-level I/O Buffer

- buffer cache in the main memory
- track cache on the disk controller



Delayed Writes and Data Loss at Machine Crash

- Writes are commonly delayed for better performance
  - data to be written is cached
- A sudden machine crash may result in a loss of data
  - a completed write does not mean the data is safely stored on storage
- fsync() – flush all delayed writes to disk
  - fsync() may not even be totally safe with delayed writes on disk controller buffer cache

Consistency: Weaker Form of Reliability

- File system operations are not atomic; a sudden machine crash may leave the file system in an inconsistent state
- Consistency checking and fix
  - it takes too long
- Journaling file system:
  - maintain a dedicated journal that logs all operations
  - the logging happens before the real operation
  - each logging is made to be atomic
  - after the completion of an operation, its entry is removed from the journal
  - at the recovery time, only journal entries need to be examined ⇒ fast recovery
  - similar to transactions in database systems
  - performance impact?

File Systems You Heard Of

- FAT
- NTFS
- ext2
- ext3
- ReiserFS

Log-Structured File Systems

- With CPUs faster, memory larger
  - buffer caches can also be larger
  - most of read requests can come from the memory cache
  - thus, most disk accesses will be writes
  - poor disk performance when most writes are small
- LFS Strategy [Rosenblum&Ousterhout SOSP1991]
  - structures entire disk as a log
  - always write to the end of the disk log
  - when updates are needed, simply add new copies with updated content; old copies of the blocks are still in the earlier portion of the log
  - periodically purge out useless blocks

Disclaimer

- Parts of the lecture slides contain original work of Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Andrew S. Tanenbaum, and Gary Nutt. The slides are intended for the sole purpose of instruction of operating systems at the University of Rochester. All copyrighted materials belong to their original owner(s).