File Buffer Cache

Applications exhibit significant locality for reading and writing files

- → Idea: cache file blocks in memory to capture locality Called the file buffer cache
 - Cache is system wide, used and shared by all processes
 - Reading from the cache makes a disk perform like memory
 - Even a small cache can be very effective

Issues

- The file buffer cache competes with VM (tradeoff here)
- Like VM, it has limited size
- Need replacement algorithms again (LRU usually used)

Caching Writes

On a write, some applications assume that data makes it through the buffer cache and onto the disk

- → As a result, writes are often slow even with caching
- OSes typically do write back caching
 - Maintain a queue of uncommitted blocks
 - Periodically flush the queue to disk (30 second threshold)
 - If blocks changed many times in 30 secs, only need one I/O
 - If blocks deleted before 30 secs (e.g., /tmp), no I/Os needed
- ✓ Unreliable, but practical
 - On a crash, all writes within last 30 secs are lost
 - Modern OSes do this by default; too slow otherwise
 - System calls (Unix: fsync) enable apps to force data to disk