23 Unix Fast File System (FFS)

Disks are statically subdivided into a number of *partitions*. Each partition may contain one *file system*. A file system consists of all the information relating to a collection of files.

File descriptors are called *i-nodes* in Unix. We've seen their structure (multi-level indexed).

A file system contains a *superblock* at a well-known block # that contains basic parameters:

- number of data blocks
- number of i-nodes (determines maximal number of files)
  - space for i-nodes is statically allocated. Typically, one i-node per 2048 bytes of disk space (assumed average file size)
- information describing the *block groups*
- disk characteristics

The disk blocks of each file systems are statically partitioned into a number of block groups, each consisting of a set of adjacent blocks (within a set of adjacent cylinders). Each block group contains:

- a replicated copy of the superblock
- space for a free block bitmap
- space for i-nodes
- a bit map of available data blocks
- data blocks.

Disk allocation heuristics try to allocate the i-node, indirect blocks, and data blocks of a file within a single blocks group, whenever possible. Moreover, a director and its files are normally allocated in the same block group. This eliminates long seeks during accesses to a single file, and a directory and its files.

Disk block allocation within a cylinder takes into account the rotational position of a disk block. When allocating a new data block to a file, try to choose a free block with an optimal rotational position, relative to the file's previous data block.

To avoid bad performance due to fragmentation, FFS keeps a reserve of 10% of the disk space. Once the disk is 90% full, normal programs can no longer create new files or extend existing files, and FFS will report that the disk is full. However, system admins (superusers) can still allocate space from the reserve.

24 ZFS

A modern, open-source filesystem developed by Sun Microsystems (now Oracle). Features:

- filesystems can span multiple partitions and block devices
- copy-on-write
- snapshots and clones (writable snapshots)
- compression, encryption, checksumming and deduplication
- replication of data and metadata blocks
- striping: writing in parallel to multiple block devices for throughput
- scrubbing: background task to detect (and repair if redundant) corrupted data and metadata
- support for very large files (2^{64} bytes) and storage pools (2^{78} bytes)
- free space is maintained as an AVL tree of extents, one per block group
- batched updates
- use of a log for metadata updates (we'll learn about this later)