NFS <-> Filesystem Communications

Helping the NFS Server suck less

O_NFSD

- Filesystems do not know operations are coming from the NFS server, so can't perform optimisations that would benefit NFSD
- Open flag would allow the caller to be easily identified.
  - cope with OOO write patterns
  - no truncate of speculative preallocation on ->release

How to write efficiently?

- what are the optimal parameters to write to the filesystem?
  - min/max write sizes
  - optimal size (e.g. sunit)
  - alignment:
    - memory buffer (e.g. needs to be page aligned)
    - file offset
    - block offset
- VFS level XFS_IOC_DIOINFO equivalent?

Client side ENOSPC detection

- Sucks – totally async, guarantees data loss
- Need to prevent overcommit caused by client side caching
- Reservation pool in NFS server?
  - per client
  - guarantees client a certain amount of space
  - requires FS to mark blocks “used” but without allocation and to use that pool on write – “alloc context”?
  - NFSv4 extension to co-ordinate client/server?

Generic VFS interfaces

- FIEMAP for efficiently discovering offline blocks (proactive NFSERR_JUKEBOX)
- generic inode flags
  - e.g. immutable, append
  - get + set operations needed
  - xattr interface would allow client side access to get+set of flags
    - need per-sb capability mask for supported flags
- ->readdirplus()

Pre-unmount notifiers

- Needed for an effective Open File Cache implementation
  - export table keeps dentry+vfsmnt references on the mntpt
  - cannot unmount due to open file reference