CS 5600 Computer Systems

Lecture 10 Part 1

Date: 11/17/2010

File Systems

Efficiency: Fraction of space storing useful content

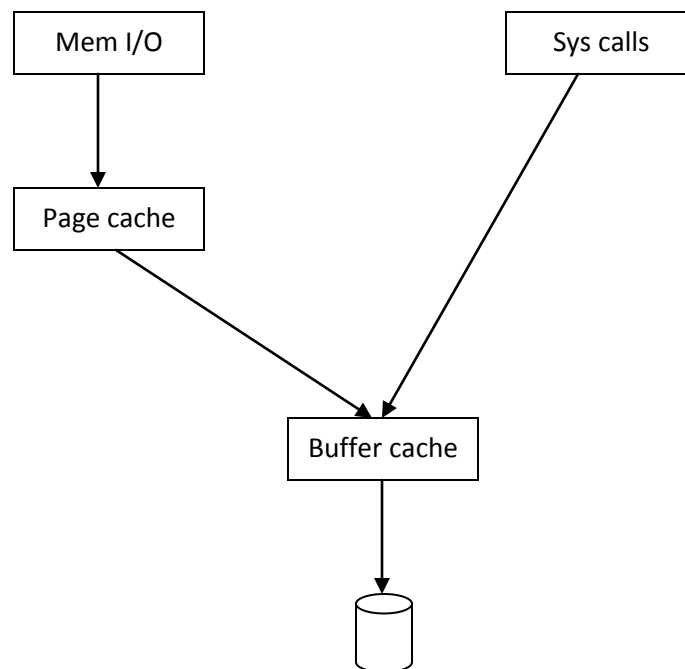Performance: Fraction of useful work to total work

E.g.:

- Disk seeks for pointer choosing
- Writing data in multi locations
- Writing/reading metadata

Add a cache to improve performance

Cache disk blocks in memory called buffer cache

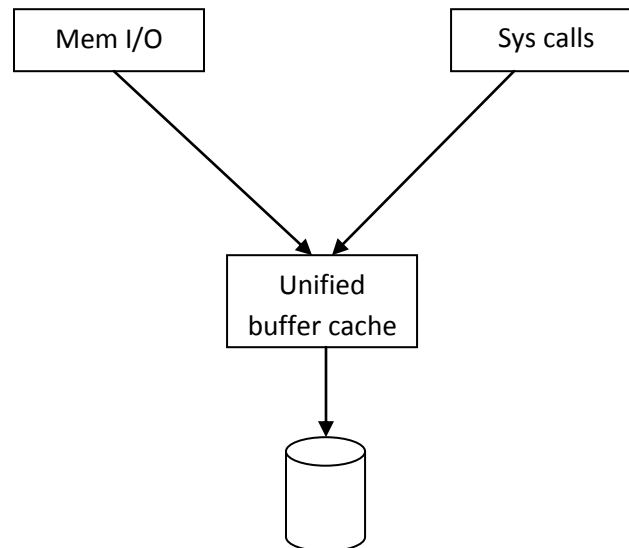2 ways of accessing the file system

1. System calls
2. Memory mapped I/O

It is bad because

- Will have to update all the cache at all time for consistency
- Wastage of memory

Resolution- unified buffer cache

```
┌─────────────┐         ┌─────────────┐
│   Mem I/O   │         │  Sys calls  │
└─────────────┘         └─────────────┘
        ╲                     ╱
         ╲                   ╱
          ▼                 ▼
        ┌───────────────────┐
        │      Unified      │
        │   buffer cache    │
        └───────────────────┘
                  │
                  ▼
                ┌───┐
                │   │
                └───┘
```

Pre-fetching disk block:

- Fetch more than 1 disk block, put into cache

Anticipate scheduling:

- Wait a small amount of time after a disk read for future requests

Remote Access of files:

 NFS- Network File System

Introduced by Sun Microsystems in 80s

Goal': Allow one machine to share with another

- Based on Client – Server model.
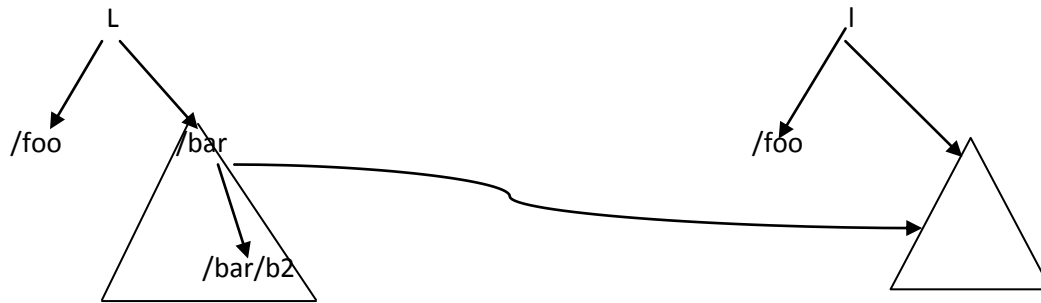- On the server, configure exports.

Each tree contains directory, {clients}, node {ro/rw}.

Authentication at NFS level is only based on IP/DNS

On the client side, the remote directory is mounted over a directory of a local file system.

Server   Client

L   I

/foo   /bar   /foo

/bar/b2

NFS implementation:

NFS is Stateless till Version 3. Stateless meaning there is no context in the communication between client and server.

Mmap, open , close are all not represented in NFS protocol.

Write() message -> data written to disk at server

NFS provides no file locking