Today's Objectives

- Priya's Talk
- Distributed File Systems
- Final Projects

Review: Priya's Talk

- "instrumented"

Review

- What is the motivation for a distributed file system?
- What does it mean for a file system to be distributed?

Review: Unix File System Operations

- `open(name, mode)`
- `creat(name, mode)`

} Return file descriptor

creat() is equivalent to open() with flags equal to O_CREAT|O_WRONLY|O_TRUNC.

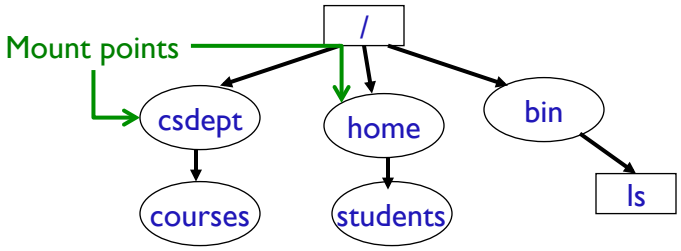
This interface is made obsolete by: open(2).

The creat() function is the same as:
open(path, O_CREAT | O_TRUNC | O_WRONLY, mode);

Distributed File System Requirements

- Transparency
  - Access, location, mobility, performance, scaling
- Concurrent file updates
- File replication
- Hardware and OS heterogeniety
- Fault tolerance
- Consistency
- Security
- Efficiency

Distributed File System Structure



- Perform ***mount*** operation to attach remote file system into local namespace
  - E.g., `/home/students` is actually a file on remote machine
  - Maps to `hydros.cs.wlu.edu:/exports/home/students`
- ***Mounting*** helps to combine files/directories in different systems and form a single file system structure

How This Looks: df

df - report file system disk space usage

```
sprenkle@python:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        7.8G     0  7.8G   0% /dev
tmpfs           7.8G   27M  7.8G   1% /dev/shm
tmpfs           7.8G  2.4M  7.8G   1% /run
tmpfs           7.8G     0  7.8G   0% /sys/fs/cgroup
/dev/sdb4       102G   15G   82G  16% /
tmpfs           7.8G  220K  7.8G   1% /tmp
/dev/sdb2       976M  169M  740M  19% /boot
/dev/sdb1       200M   16M  185M   8% /boot/efi
hydros:/home    493G  248G  220G  53% /home
tmpfs           1.6G   20K  1.6G   1% /run/user/42
tmpfs           1.6G  6.2M  1.6G   1% /run/user/1677
tmpfs           1.6G  3.8M  1.6G   1% /run/user/1161
tmpfs           1.6G     0  1.6G   0% /run/user/1501
```

No one had accessed /home/courses on python since last reboot

How This Looks: df

df - report file system disk space usage

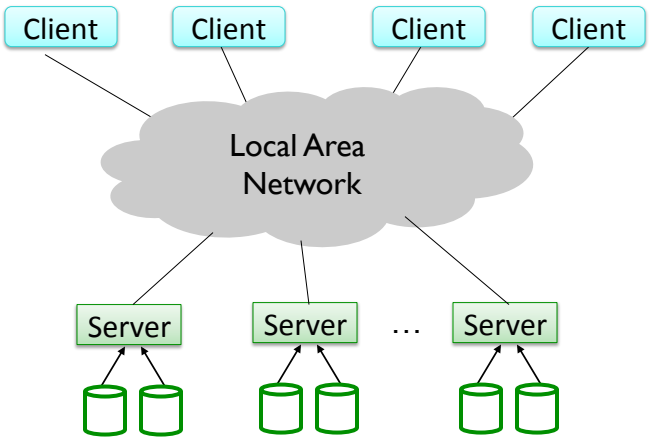
```
sprenkle@python:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        7.8G     0  7.8G   0% /dev
tmpfs           7.8G   27M  7.8G   1% /dev/shm
tmpfs           7.8G  2.4M  7.8G   1% /run
tmpfs           7.8G     0  7.8G   0% /sys/fs/cgroup
/dev/sdb4       102G   15G   82G  16% /
tmpfs           7.8G  220K  7.8G   1% /tmp
/dev/sdb2       976M  169M  740M  19% /boot
/dev/sdb1       200M   16M  185M   8% /boot/efi
hydros:/home    493G  248G  220G  53% /home
hydros:/local    99G  9.0G   85G  10% /csdept
tmpfs           1.6G   20K  1.6G   1% /run/user/42
tmpfs           1.6G  6.2M  1.6G   1% /run/user/1677
tmpfs           1.6G  3.8M  1.6G   1% /run/user/1161
tmpfs           1.6G     0  1.6G   0% /run/user/1501
```

Design Issue: Naming

- Based on the naming structure, you can't tell whether a file is local or remote
  - `/home/students/yourusername`
  - Vs
  - `/usr/bin/ls`
- Another example: using ~username to refer to user's directory
  - Could be mapped to /home/students/username in client 1 and to /usr/students/username in client 2
- Result: Location Transparency

DFS: Performance

- How can we make distributed file access approximate the performance of local file access?



DFS: Performance

- How to make distributed file access approximate the performance of local file access?
  - Caching: take advantage of locality
    - Both *spatial* and *temporal*

DFS: Performance

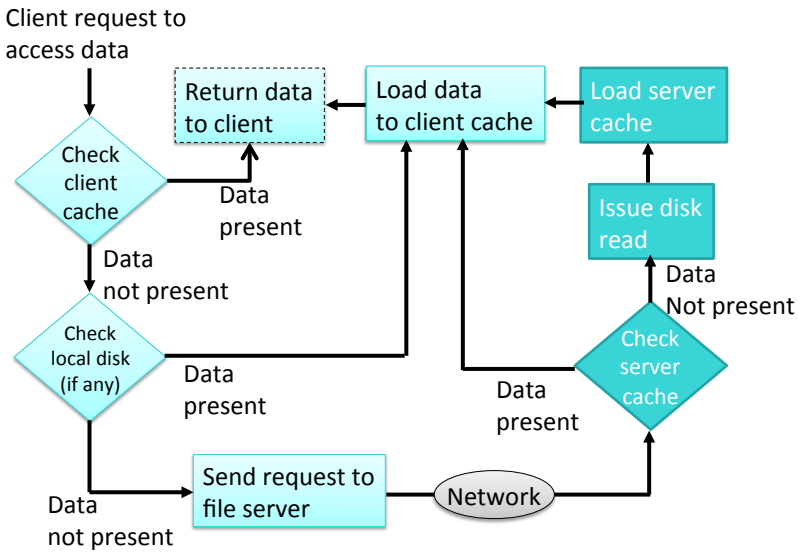
- How to make distributed file access approximate the performance of local file access?
- Caching: take advantage of locality
  - ***Temporal locality***: something that is accessed will likely be accessed in the near future
  - ***Spatial Locality****:* a resource is more likely to be accessed if something "near" it was recently accessed

What issues are introduced by caching?

UNIX File Usage

- Most files are small (< 10k)
- Reads outnumber writes (~6:1)
- Sequential access is common
  - (example of ***spatial locality***)
- Files remain open for short period of time
  - 75% < .5s, 90% < 10s
- Most files accessed by exactly one user
  - Most shared files written by exactly one user
- ***Temporal locality***: recently accessed files likely to be accessed again in near future
- Most bytes/files are short lived

DFS Data Access



Final Project Discussion

Looking Ahead

- Use next Wednesday or Friday for Exam?