CS615 - Aspects of System Administration

Filesystems, Disks, Storage

Department of Computer Science
Stevens Institute of Technology
Jan Schaumann
jschauma@stevens.edu
http://www.cs.stevens.edu/~jschauma/615A/

Let's review HW1

Running an instance:

```
$ aws ec2 run-instances --key-name stevens --image-id <AMI-ID>
```

Let's review HW1

Save yourself some typing:

```
$ alias instance='aws ec2 run-instances --key-name stevens \
                        --image-id'
$ instance <AMI-ID>
```

Let's review HW1

Make it permanent:

```
$ echo "alias instance='aws ec2 run-instances --key-name stevens \
        --image-id'" >> ~/.bashrc
$ . ~/.bashrc
$ alias
alias instance='aws ec2 run-instances --key-name stevens --image-id'
$
```

Let's review HW1

Start an OmniOS instance:

```
$ aws ec2 describe-images --query Images[].ImageId --output text \
        --filters "Name=name,Values='OmniOS r151014 LTS'"
ami-9fbbfaf5
$ instance ami-9fbbfaf5
[...]
```

Let's review HW1

ssh to an instance:

```
$ ssh -i ~/.ssh/ec2 root@<mumble>.compute-1.amazonaws.com
```

Let's review HW1

Let's save ourselves some typing:

```
$ cat >>~/.ssh/config <<EOF
> Host *.amazonaws.com
>         IdentityFile ~/.ssh/ec2
>         User         root
> EOF
$ ssh <mumble>.compute-1.amazonaws.com
```

Let's review HW1

How do we know what host we're connecting to?

```
$ ssh ec2-174-129-71-111.compute-1.amazonaws.com
The authenticity of host 'ec2-174-129-71-111.compute-1.amazonaws.com
(174.129.71.111)' can't be established.
RSA key fingerprint is
d2:20:0f:38:11:37:a0:1c:ca:7a:9b:cb:11:28:0b:f7.
Are you sure you want to continue connecting (yes/no)?
```

Let's review HW1

```
$ instance ami-0187f76b # Fedora
$ aws ec2 get-console-output --instance-id i-18fd6d98
[...]
<14>Feb  6 18:37:59 ec2: ############################################################
<14>Feb  6 18:37:59 ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----
<14>Feb  6 18:37:59 ec2: 256 SHA256:3kQfQTV+sfk/BDa9jGdvvea8i28TwWOgWt8lZKdALgA /etc/
<14>Feb  6 18:37:59 ec2: 256 SHA256:R3/H4kFMu/JZZCxQL7ZYODZoKwGFNBN6Cpfbsx/PFDQ /etc/
<14>Feb  6 18:37:59 ec2: 2048 SHA256:WjIFQE4ZEQfSR0bfCVgRbR7Q0780RFug8CZNXcEam0I /etc
<14>Feb  6 18:37:59 ec2: -----END SSH HOST KEY FINGERPRINTS-----
-----BEGIN SSH HOST KEY KEYS-----
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBP4RT8CDVexvk
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBTpcfKZcASSjicvMPAx6UAQSac9ZH0UeO+/A9m6EuFO
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC48feKNKyaS+c+zJmu6/L6o9L7VkW6YMuwu0//Uo5uS46Yk
-----END SSH HOST KEY KEYS-----
```

Let's review HW1

How do we know what host we're connecting to?

```
$ instance ami-0187f76b # Fedora
$ aws ec2 get-console-output --instance-id i-18fd6d98 | grep ^ecdsa > hostkey
$ ssh-keygen -l -f hostkey
256 85:24:1d:56:d7:b8:9e:27:12:61:94:4a:72:df:d0:9a (ECDSA)
$  ssh fedora@ec2-54-204-86-158.compute-1.amazonaws.com
The authenticity of host 'ec2-54-204-86-158.compute-1.amazonaws.com
(54.204.86.158)' can't be established.
ECDSA key fingerprint is 85:24:1d:56:d7:b8:9e:27:12:61:94:4a:72:df:d0:9a.
Are you sure you want to continue connecting (yes/no)? yes
[fedora@ip-10-170-2-33 ~]$
```

Let's review HW1

```
# uname -a
SunOS ip-10-152-178-106.ec2.internal 5.11 omnios-8322307 i86pc i386 i86xpv
```

Let's review HW1

```
# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
xnf0: flags=1004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4> mtu 1500 index 2
        inet 10.235.81.115 netmask ffffffc0 broadcast 10.235.81.127
        ether 22:0:a:eb:51:73
lo0: flags=2002000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv6,VIRTUAL> mtu 8252 index 1
        inet6 ::1/128
xnf0: flags=20002000840<RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        inet6 ::/0
        ether 22:0:a:eb:51:73
```

Let's review HW1

```
# netstat -na | more
[...]
TCP: IPv4
   Local Address        Remote Address    Swind Send-Q Rwind Recv-Q
State
-------------------- -------------------- ----- ------ ----- ------ -----------
127.0.0.1.4999             *.*                0      0 128000      0 LISTEN
      *.111                *.*                0      0 128000      0 LISTEN
      *.*                  *.*                0      0 128000      0 IDLE
      *.111                *.*                0      0 128000      0 LISTEN
      *.*                  *.*                0      0 128000      0 IDLE
      *.46457              *.*                0      0 128000      0 LISTEN
      *.55986              *.*                0      0 128000      0 LISTEN
      *.22                 *.*                0      0 128000      0 LISTEN
10.110.94.225.22     155.246.89.107.46137 42304     47 128592      0 ESTABLISHED
[...]
```

Let's review HW1

```
# man df
[...]
# df
[...]
# df -haT
[...]
# df -i
[...]
# mount
[...]
```

Let's review HW1

```
# format
[...]
format> verify
[...]
# zpool list
[...]
# zfs list
[...]
```

Filesystems, Disks, Storage

```
$ ssh linux-lab.cs.stevens.edu
$ dh -hT
```

How much disk space is available under `/tmp`?

Filesystems, Disks, Storage

```
$ ssh linux-lab.cs.stevens.edu
$ dh -hT
$ cd /tmp
$ df -h .
```

How much disk space is available under `/tmp`?

Fill it up.

Filesystems, Disks, Storage

File sizes are not always what they seem to be.

```
$ mkdir /tmp/${USER}
$ export LARGE=/tmp/${USER}/large
$ truncate -s $(df /tmp | awk '/^\// { print $4; }')0000 ${LARGE}
$ ls -l ${LARGE}
$ du ${LARGE}
$ stat ${LARGE}
$ cp ${LARGE} ${LARGE}2
$ du ${LARGE}2
$ cat ${LARGE} > ${LARGE}2
$ ls -l ${LARGE}*
$ du ${LARGE}*
```

Filesystems, Disks, Storage

How many files can be created on `/tmp`?

```
$ ssh linux-lab.cs.stevens.edu
$ rm /tmp/${USER}/large2
$ dh -i /tmp
```

Filesystems, Disks, Storage

How many files can be created on `/tmp`?

```
$ ssh linux-lab.cs.stevens.edu
$ dh -i /tmp
```

Fill 'em up!

Filesystems, Disks, Storage

Error messages aren't always what they seem to be!

```
$ ssh linux-lab.cs.stevens.edu
$ dh -i /tmp
$ echo hello >/tmp/${USER}/hello
$ cc -Wall ~jschauma/tmp/mkfiles.c
$ ./a.out
$ ls -ld /tmp/$(id -u)
$ ls /tmp/$(id -u) | wc -l
$ touch /tmp/${USER}/newfile
$ echo "hello hello hello" >> /tmp/${USER}/hello
$ rm -fr /tmp/$(id -u)
```

Filesystems, Disks, Storage

Adding, (re-)partitioning, mounting disks requires understanding of:

- basic disk concepts
- basic filesystem concepts
- file systems

Topics covered

Adding, (re-)partitioning, mounting disks requires understanding of:

- basic disk concepts
  - storage models
  - disk interfaces
  - physical disk structure
  - partitions
- basic filesystem concepts
- file systems

Topics covered

Adding, (re-)partitioning, mounting disks requires understanding of:

- basic disk concepts
  - storage models
  - disk interfaces
  - physical disk structure
  - partitions
- basic filesystem concepts
  - RAID
  - logical volume managment
  - device formatting
- file systems

Topics covered

Adding, (re-)partitioning, mounting disks requires understanding of:

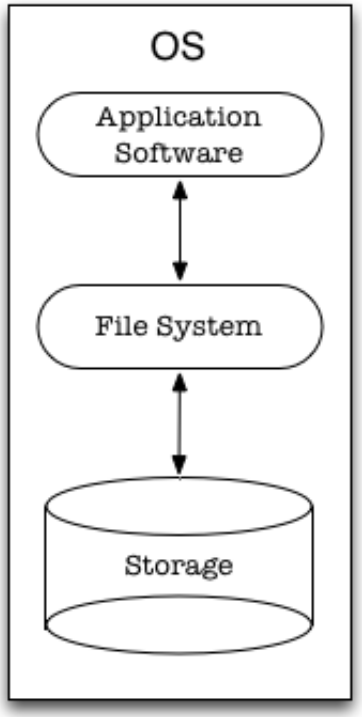
- basic disk concepts
  - storage models
  - disk interfaces
  - physical disk structure
  - partitions
- basic filesystem concepts
  - RAID
  - logical volume managment
  - device formatting
- file systems
  - the UNIX filesystem or Berkeley Fast File System (FFS)

Basic Disk Concepts

Storage Models

Basic Disk Concepts: Storage Models

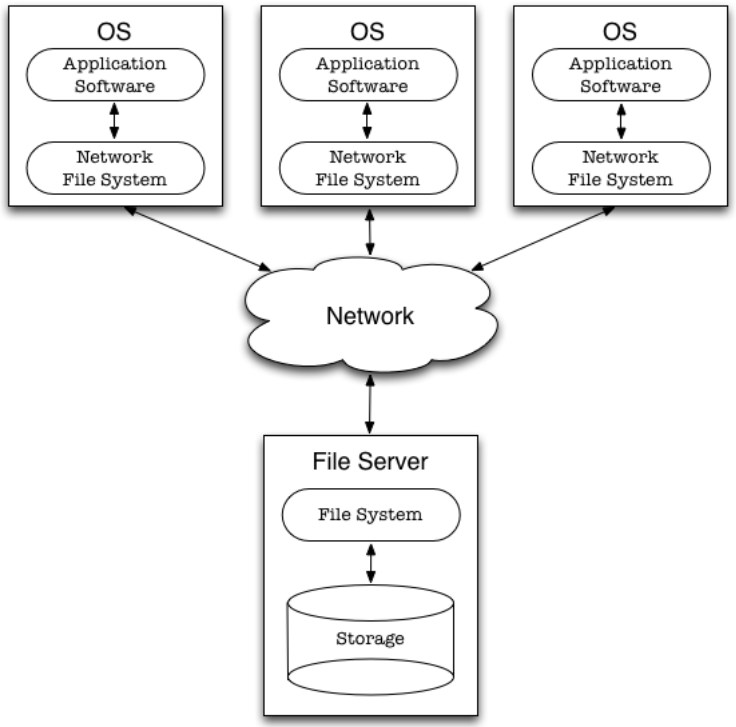
Direct Attached Storage (DAS)




```
ssh lab 'df -hT /'
```

Basic Disk Concepts: Storage Models

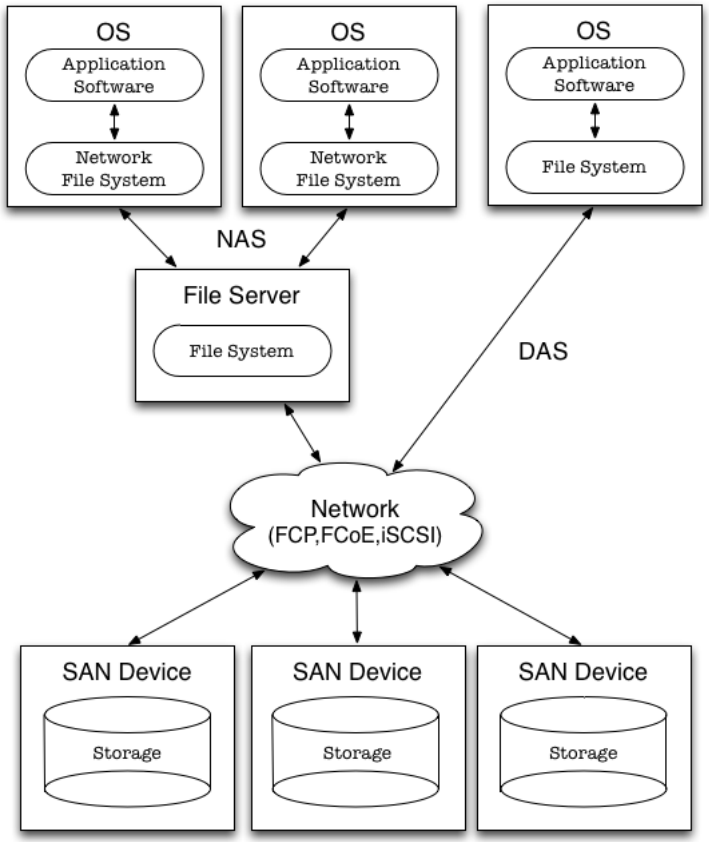
Network Attached Storage (NAS)




```
ssh lab 'df -hT /home/$(whoami)'
```

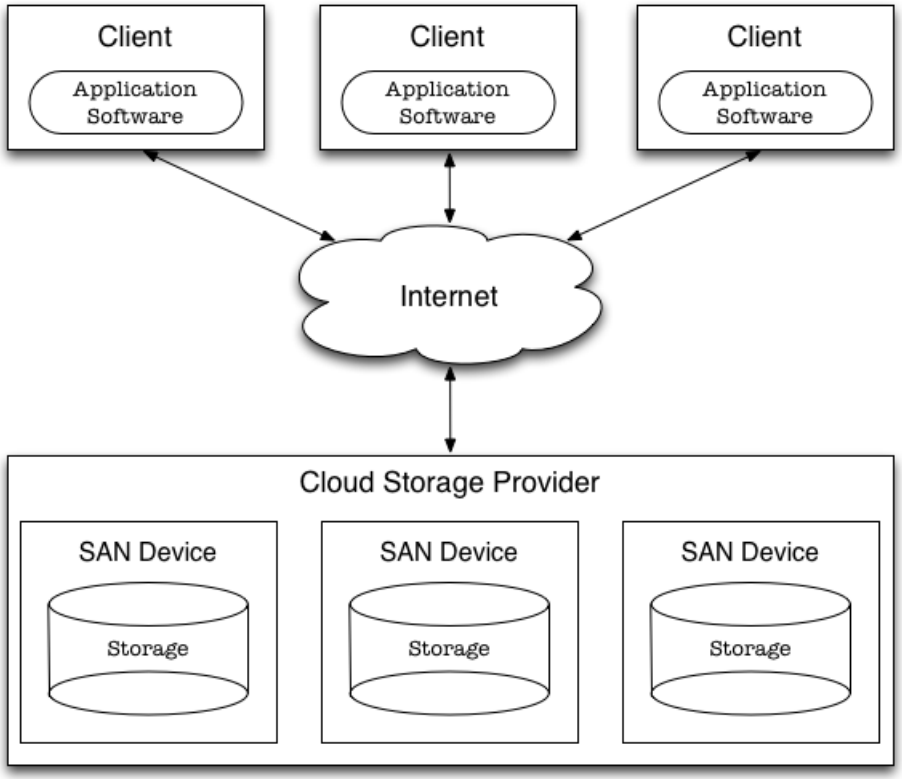
Basic Disk Concepts: Storage Models

Storage Area Networks (SAN)



Basic Disk Concepts: Storage Models

Cloud Storage (Examples: EBS, S3)



Basic Disk Concepts

Disk Devices

Basic Disk Concepts: Disk Devices

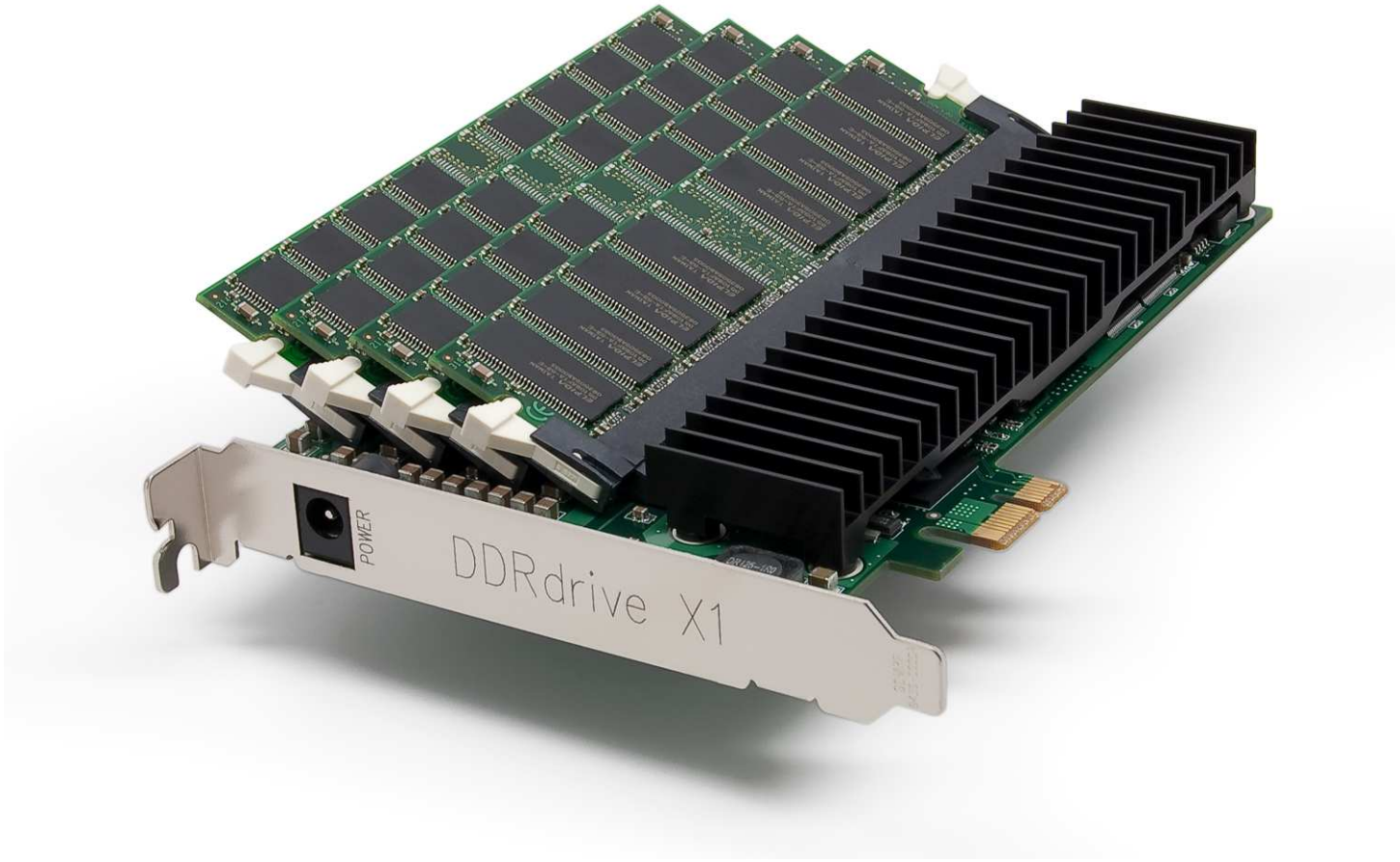
Basic Disk Concepts: Disk Devices

Security affects everything.

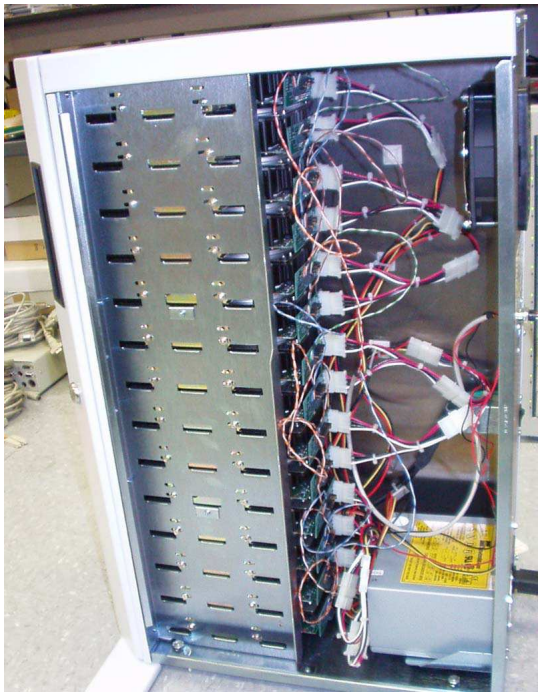
`http://t.co/eM6XpATITQ`

Basic Disk Concepts: Disk Devices

Basic Disk Concepts: Disk Devices



Basic Disk Concepts: Disk Devices



Basic Disk Concepts: Disk Devices

Basic Disk Concepts: Disk Devices

Basic Disk Concepts

Disk Interfaces

Basic Disk Concepts: Disk Interfaces: SCSI

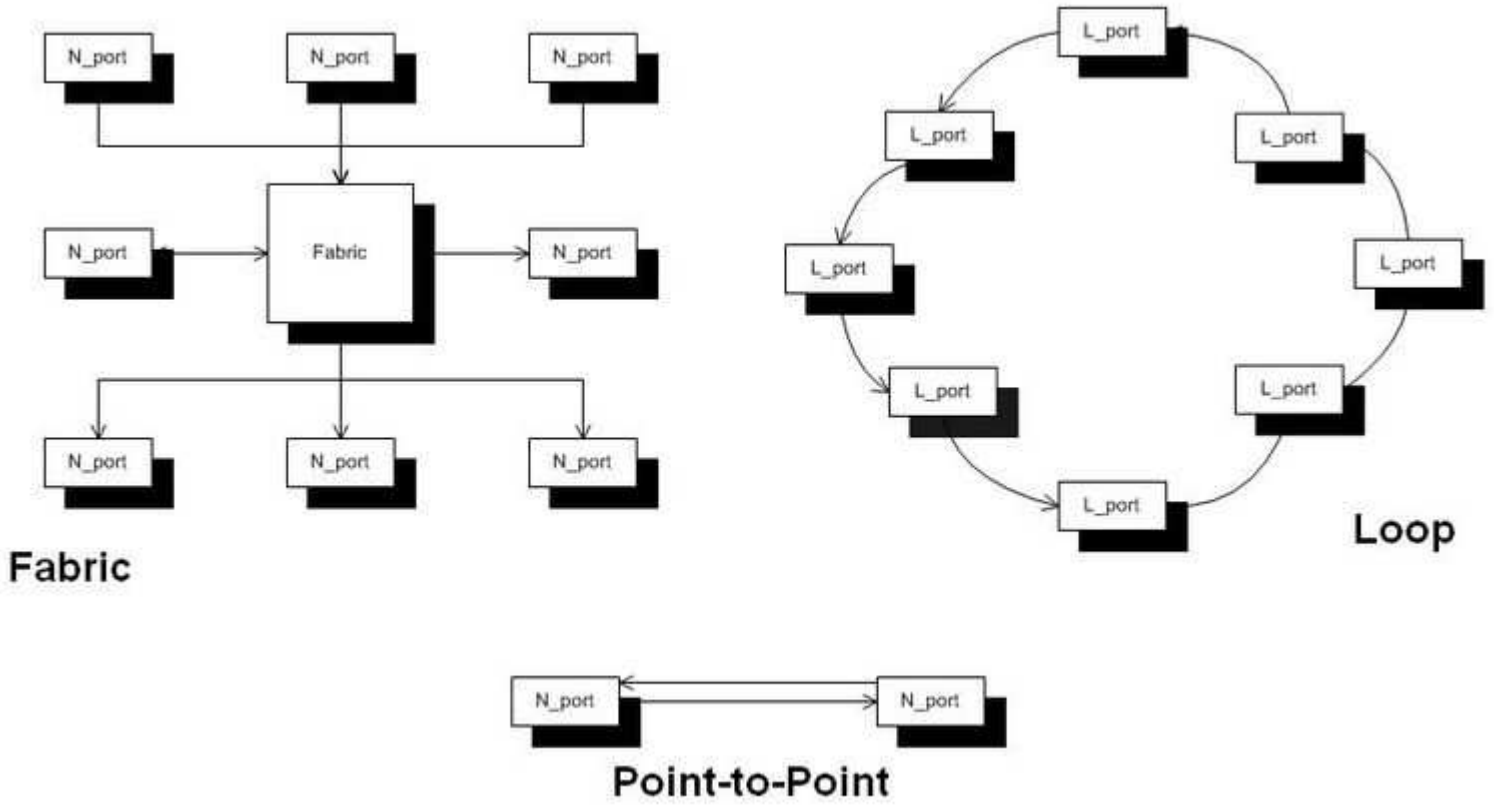
Basic Disk Concepts: Disk Interfaces: ATA

Basic Disk Concepts: Disk Interfaces: ATA

Basic Disk Concepts: Disk Interfaces: Fibre Channel

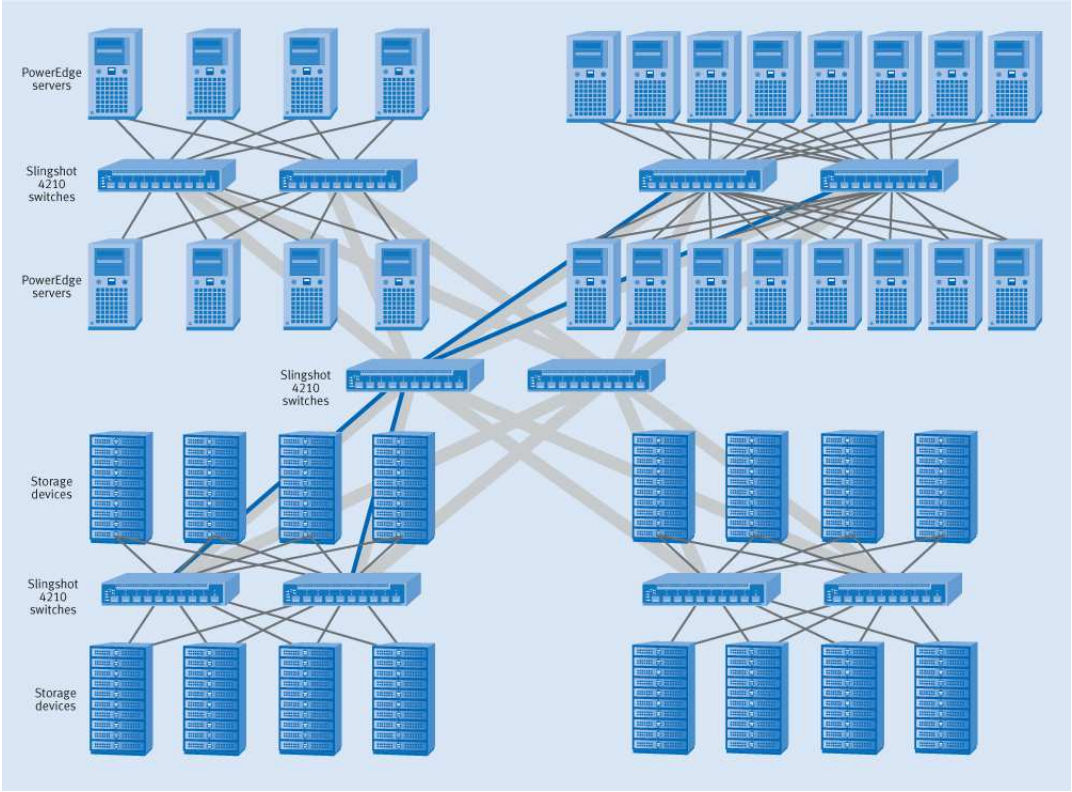
Basic Disk Concepts: Disk Interfaces: Fibre Channel

Basic Disk Concepts: Disk Interfaces: Fibre Channel



Basic Disk Concepts: Disk Interfaces: Fibre Channel

Basic Disk Concepts: Disk Interfaces: Fibre Channel



Basic Disk Concepts: Disk Interfaces: SANs

- ATA over Ethernet (*AoE*):
  - create low-cost SAN
  - ATA encapsulated into Ethernet frames
- Fibre Channel over Ethernet (*FCoE*):
  - consolidate IP and FC/SAN networks
  - FC encapsulated into Ethernet frames
- *oE:
  - no TCP/IP overhead
  - restricted to a single Layer 2 network
  - no inherent security features
- iSCSI
  - SCSI encapsulated in TCP/IP packets

Basic Disk Concepts

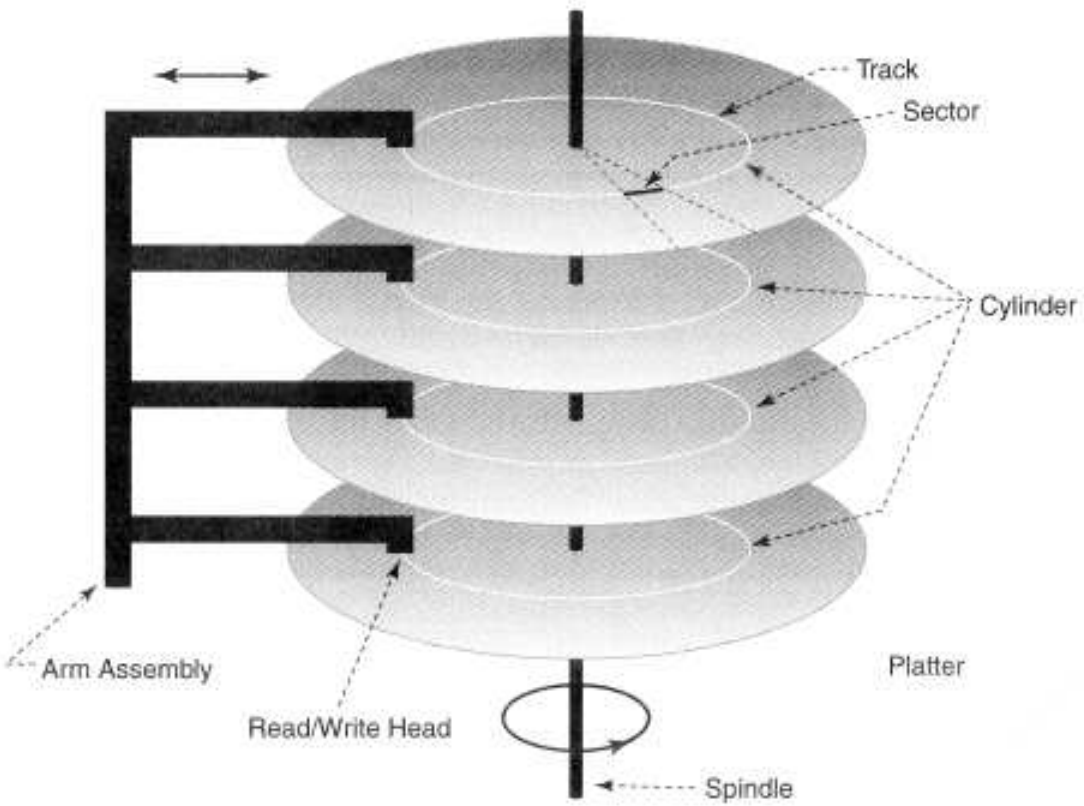
Physical Disk Structure

Basic Disk Concepts: Disk Devices

Basic Disk Concepts: Disk Devices

Basic Disk Concepts: Disk Devices

Basic Disk Concepts: Physical Disk Structure

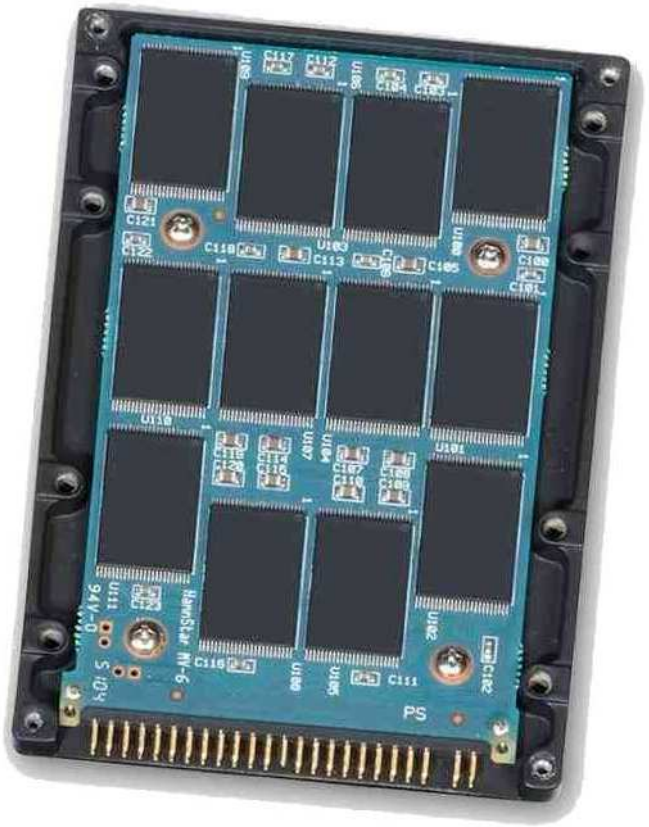


Basic Disk Concepts: Physical Disk Structure

Hard drive performance determined by:

- seek time
- rotational latency
- internal data rate
- a few other negligible factors (external data rate, command overhead, access time, etc.)

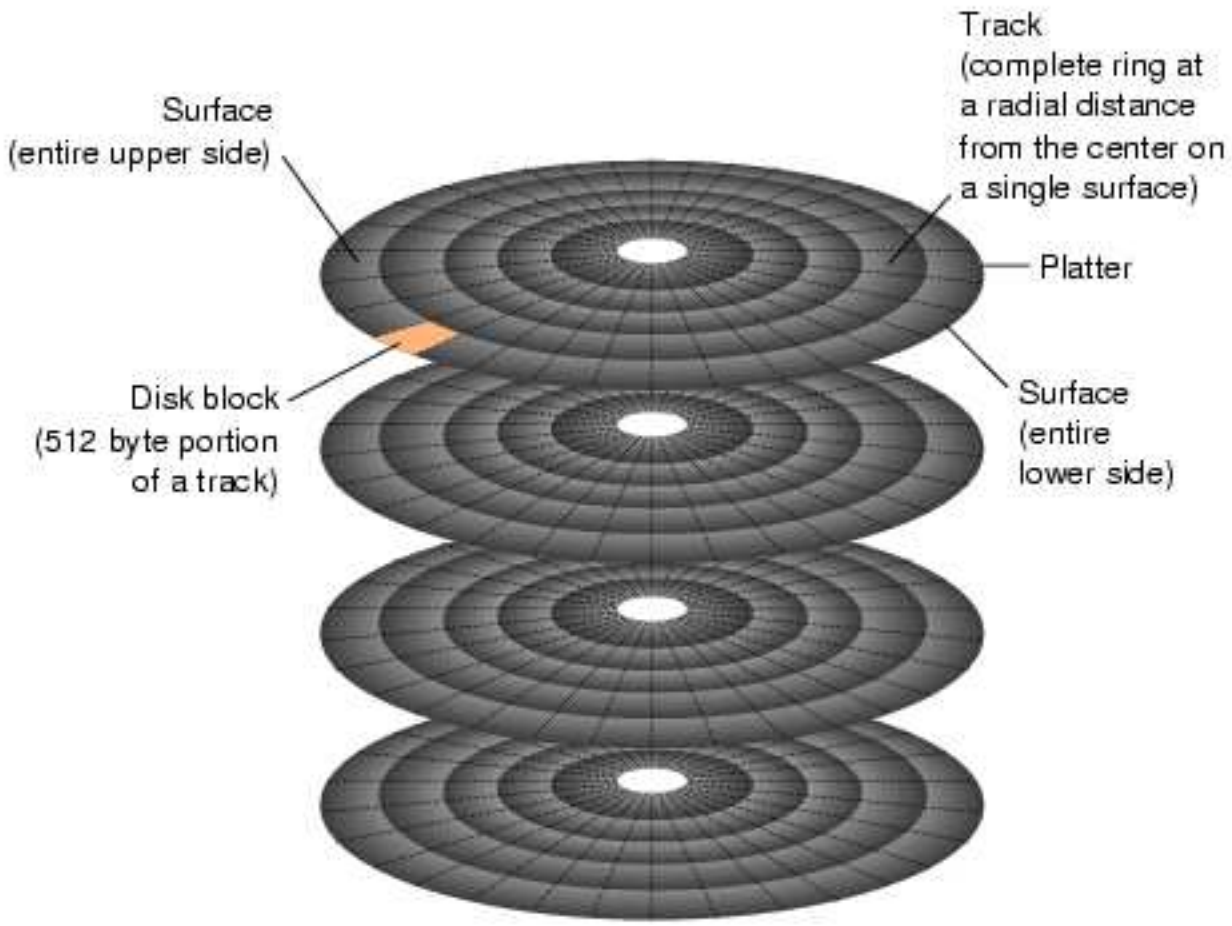
Basic Disk Concepts: Disk Devices



Basic Disk Concepts

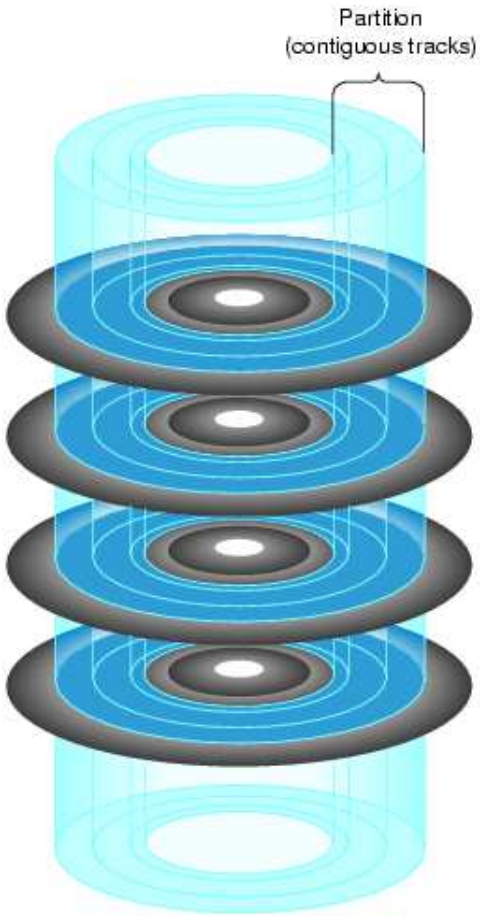
Partitions

Basic Disk Concepts: Partitions

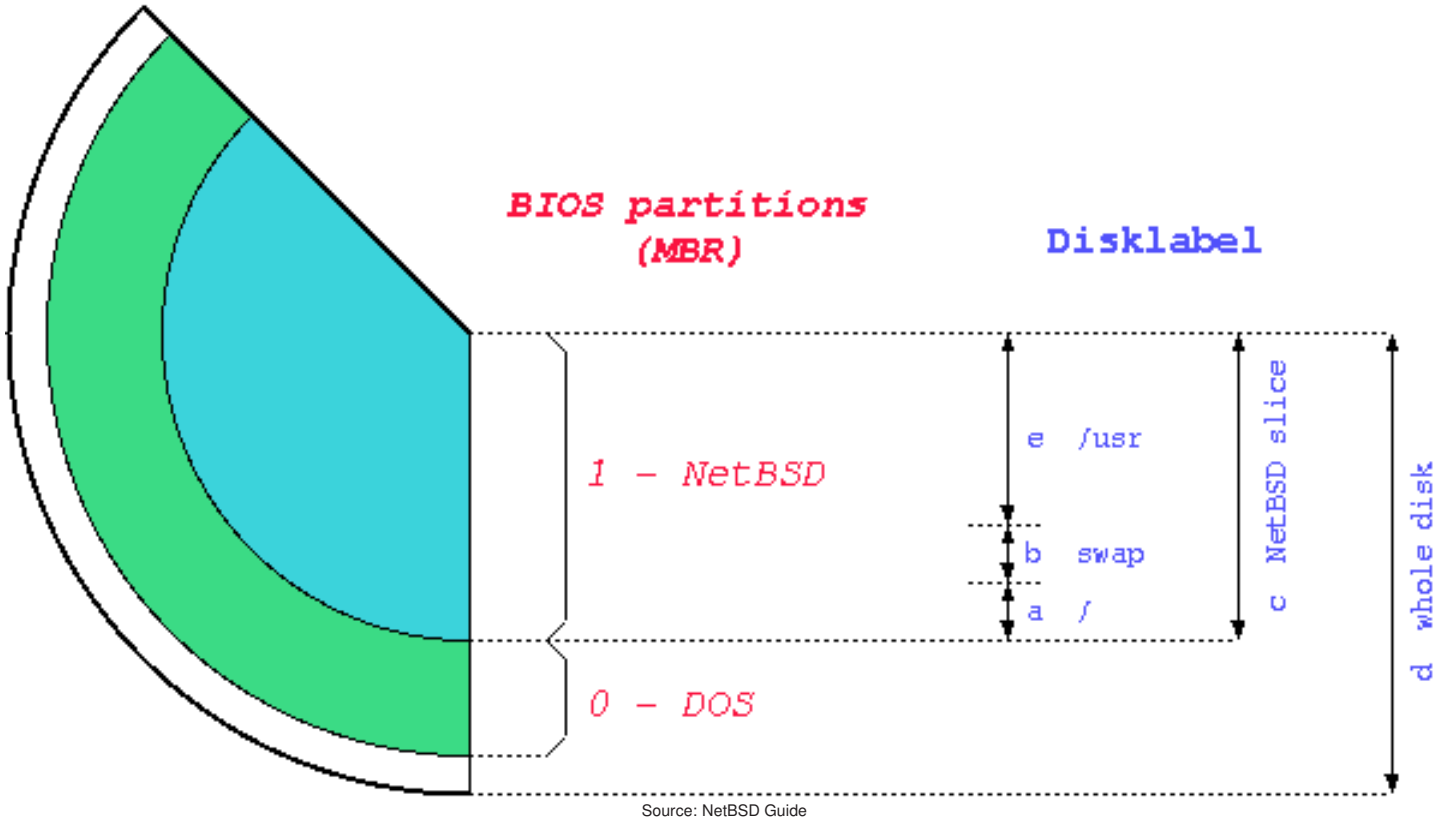


Source: SGI Techpubs

Basic Disk Concepts: Partitions



Basic Disk Concepts: Partitions



Source: NetBSD Guide

Basic Disk Concepts: Partitions

NetBSD example (from `disklabel(8)`)
Partition 'a': /
Partition 'b': swap
Partition 'e': /home

```
#        size    offset   fstype [fsize bsize cpg/sgs]
a:  20972385        63   4.2BSD   4096 32768  1180  # (Cyl.      0*- 20805)
b:   1048320  20972448     swap                      # (Cyl.  20806 - 21845)
c:  78140097        63   unused      0     0        # (Cyl.      0*- 77519)
d:  78140160         0   unused      0     0        # (Cyl.      0 - 77519)
e:  56119392  22020768   4.2BSD   4096 32768 58528  # (Cyl.  21846 - 77519)
```

Basic Disk Concepts: Partitions

NetBSD example (from `disklabel(8)`)
Partition 'a': / 10 GB
Partition 'b': swap
Partition 'e': /home 26 GB

```
#        size    offset   fstype [fsize bsize cpg/sgs]
a:   20972385        63   4.2BSD   4096 32768  1180  # (Cyl.      0*- 20805)
b:    1048320  20972448     swap                      # (Cyl.  20806 - 21845)
c:   78140097        63   unused      0     0        # (Cyl.      0*- 77519)
d:   78140160         0   unused      0     0        # (Cyl.      0 - 77519)
e:   56119392  22020768   4.2BSD   4096 32768 58528  # (Cyl.  21846 - 77519)
```

Basic Disk Concepts: Partitions

Solaris example (from `format(1m)`):

```
Current partition table (original):
Total disk cylinders available: 38758 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       3 -  3764       3.62GB    (3762/0/0)   7584192
  1       swap    wu    3765 -  4364     590.62MB    (600/0/0)    1209600
  2     backup    wm       0 - 38757      37.26GB    (38758/0/0) 78136128
  3 unassigned    wm       0                  0      (0/0/0)            0
  4 unassigned    wm       0                  0      (0/0/0)            0
  5 unassigned    wm       0                  0      (0/0/0)            0
  6 unassigned    wm       0                  0      (0/0/0)            0
  7       home    wm    4365 - 38757      33.06GB    (34393/0/0) 69336288
  8       boot    wu       0 -     0       0.98MB    (1/0/0)          2016
  9 alternates    wu       1 -     2       1.97MB    (2/0/0)          4032
```

Basic Disk Concepts: Partitions

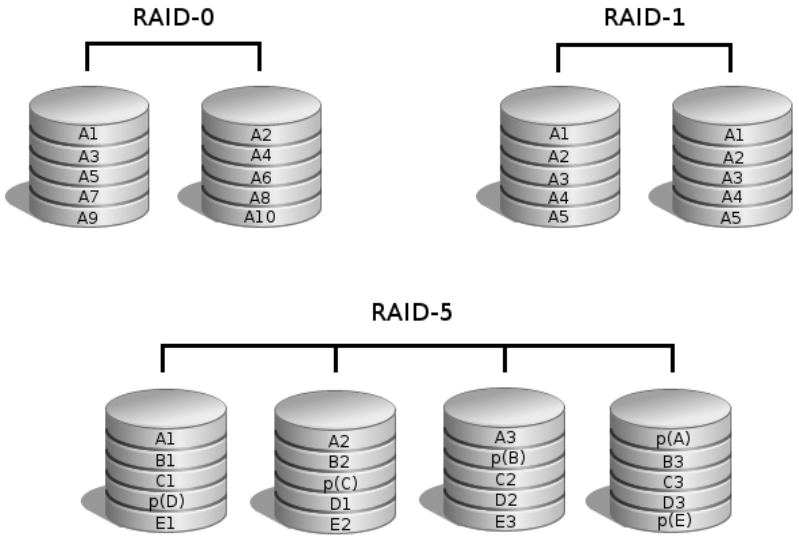
Linux example (from `fdisk(8)`):

```
Disk /dev/sda: 80.0 GB, 80000000000 bytes
255 heads, 63 sectors/track, 9726 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          33      265041   83  Linux
/dev/sda2              34        9726    77859022+  83  Linux
```

Basic Disk and Filesystem Concepts: RAID and Logical Volumes

- allow file systems to be larger than the physical size of a disk
- inrease I/O performance when *striped*
- fault tolerant when *mirrored* or *plexed*



Hooray!

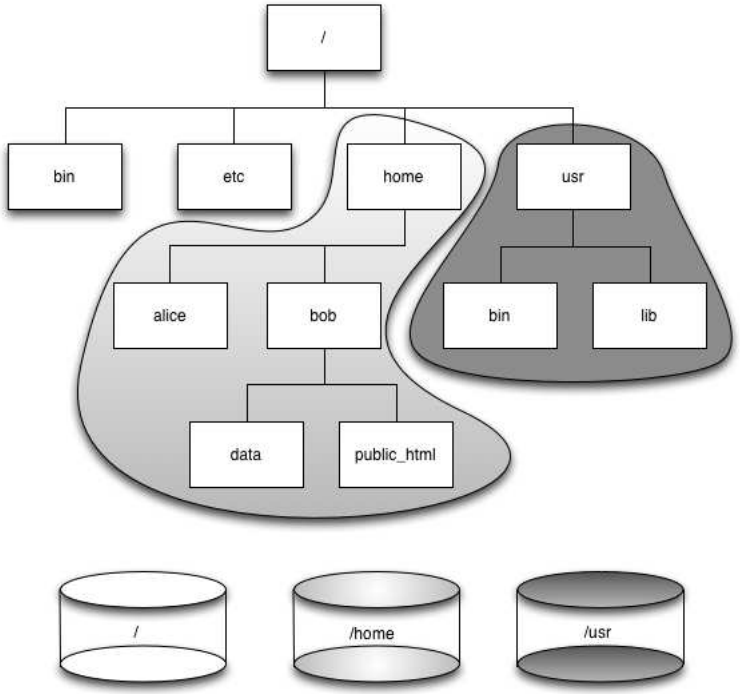
5 Minute Break

Basic Filesystem Concepts

Filesystem Layout

Basic Filesystem Concepts

All partitions – with the exception of the *root* (or /) partition – can be *mounted* anywhere in the filesystem hierarchy.



Basic Filesystem Concepts

All partitions – with the exception of the *root* (or `/`) partition – can be *mounted* anywhere in the filesystem hierarchy.

The file `/etc/fstab` (see fstab(5)) specifies which disks / partitions to mount where:

```
/dev/wd0a    /         ffs    rw 1 1
/dev/cgd1a   none      swap   sw 0 0
/dev/cgd0a   /home     ffs    rw 1 2
/ignoreme    /tmp      mfs    rw,-b4096,-f512,-s262144 0 0
kernfs       /kern     kernfs rw
procfs       /proc     procfs rw,noauto
ptyfs        /dev/pts  ptyfs  rw 0 0
```

Basic Filesystem Concepts

All partitions – with the exception of the *root* (or `/`) partition – can be *mounted* anywhere in the filesystem hierarchy.

The file `/etc/fstab` (see fstab(5)) specifies which disks / partitions to mount where:

```
# /etc/fstab: static file system information.
#
# Use 'vol_id --uuid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
# / was on /dev/sda2 during installation
LABEL=ROOT      /       ext3    errors=remount-ro,acl   0       1
# /boot was on /dev/sda1 during installation
LABEL=BOOT      /boot   ext3    defaults,acl            0       2
# swap was on /dev/sda5 during installation
UUID=9329ae83-289d-4c3d-8756-f707c4bbb312 none            swap    sw
0       0
/dev/scd0       /media/cdrom0   udf,iso9660 user,noauto,exec,utf8 0       0
deathstar.phy.stevens-tech.edu:/export/nfs-sw/opt    /opt            nfs ro,rsize=32768,intr,nolock  0 0
deathstar.phy.stevens-tech.edu:/export/srcit-dist    /mnt/srcit-dist  nfs ro,rsize=32768,intr,nolock  0 0
corsario.cs.stevens-tech.edu:/export/people          /mnt/legacy/people nfs rw,rsize=32768,wsize=32768,intr,lock   0 0
corsario.cs.stevens-tech.edu:/export/faculty         /mnt/legacy/faculty nfs rw,rsize=32768,wsize=32768,intr,lock  0 0
```

Basic Filesystem Concepts

All partitions – with the exception of the *root* (or `/`) partition – can be *mounted* anywhere in the filesystem hierarchy.

To see what filesystems are currently mounted, run `mount(8)`:

```
/dev/wd0a on / type ffs (local)
/dev/cgd0a on /home type ffs (local)
mfs:276 on /tmp type mfs (synchronous, local)
kernfs on /kern type kernfs (local)
ptyfs on /dev/pts type ptyfs (local)
```

Basic Filesystem Concepts

```
$ mount
/dev/sda2 on / type ext3 (rw,errors=remount-ro,acl)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
varrun on /var/run type tmpfs (rw,nosuid,mode=0755)
varlock on /var/lock type tmpfs (rw,noexec,nosuid,nodev,mode=1777)
udev on /dev type tmpfs (rw,mode=0755)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=620)
fusectl on /sys/fs/fuse/connections type fusectl (rw)
lrm on /lib/modules/2.6.28-17-generic/volatile type tmpfs (rw,mode=755)
/dev/sda1 on /boot type ext3 (rw,acl)
securityfs on /sys/kernel/security type securityfs (rw)
automount(pid2623) on /home type autofs (rw,fd=4,pgrp=2623,minproto=2,maxproto=4)
deathstar.phy.stevens-tech.edu:/export/nfs-sw/opt on /opt type nfs (ro,rsize=32768,intr,nolock,addr=155.246.89.4)
deathstar.phy.stevens-tech.edu:/export/srcit-dist on /mnt/srcit-dist type nfs (ro,rsize=32768,intr,nolock,addr=155.246.89.4)
corsario.cs.stevens-tech.edu:/export/people on /mnt/legacy/people type nfs (rw,rsize=32768,wsize=32768,intr,lock,addr=155.246.89.20)
corsario.cs.stevens-tech.edu:/export/faculty on /mnt/legacy/faculty type nfs (rw,rsize=32768,wsize=32768,intr,lock,addr=155.246.89.20)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,noexec,nosuid,nodev)
deathstar.phy.stevens-tech.edu:/export/home/kamberov on /home/kamberov type nfs (rw,sync,intr,vers=3,sloppy,addr=155.246.89.4)
deathstar.phy.stevens-tech.edu:/export/home/mweiss on /home/mweiss type nfs (rw,sync,intr,vers=3,sloppy,addr=155.246.89.4)
deathstar.phy.stevens-tech.edu:/export/home/jschauma on /home/jschauma type nfs (rw,sync,intr,vers=3,sloppy,addr=155.246.89.4)
```

Basic Filesystem Concepts

Some of the different kinds of filesystems:

Basic Filesystem Concepts

Some of the different kinds of filesystems:

- “Regular” File Systems
- Journaling File Systems
- Network File Systems
- Various

Basic Filesystem Concepts

The UNIX Filesystem

Basic Filesystem Concepts: The UNIX Filesystem

The filesystem is responsible for storing the data on the disk. So to read/write data, it needs to know in which physical blocks the actual data is located; ie how to map files to the disk blocks.

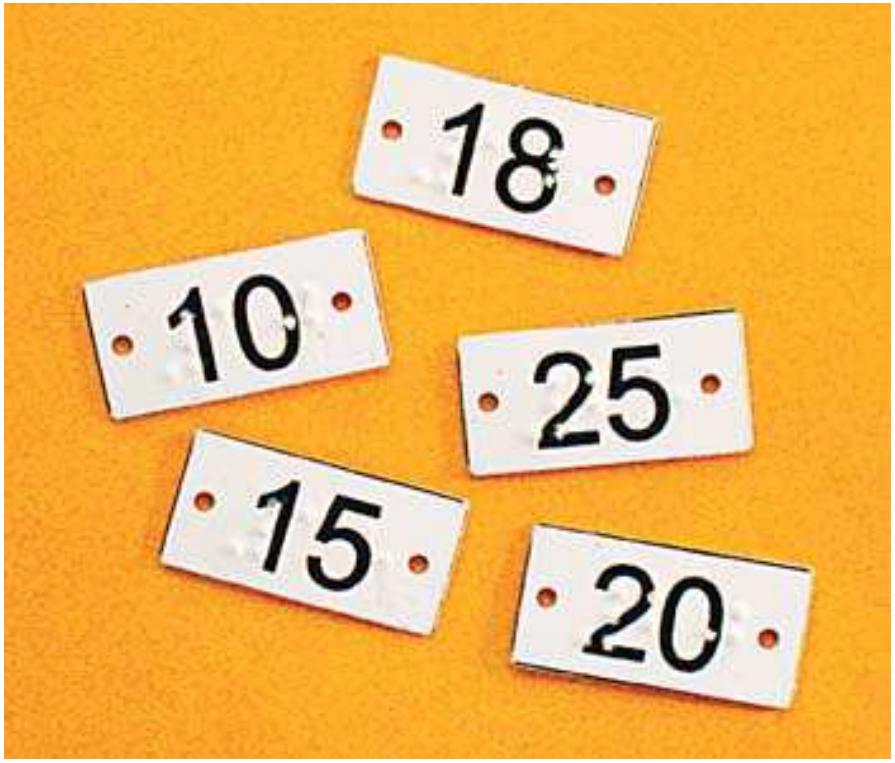
Basic Filesystem Concepts: The UNIX Filesystem

Basic Filesystem Concepts: The UNIX Filesystem

Basic Filesystem Concepts: The UNIX Filesystem

Basic Filesystem Concepts: The UNIX Filesystem

Basic Filesystem Concepts: The UNIX Filesystem



Basic Filesystem Concepts: The UNIX Filesystem

Basic Filesystem Concepts: The UNIX Filesystem

Basic Filesystem Concepts: The UNIX Filesystem

Basic Filesystem Concepts: The UNIX Filesystem

Basic Filesystem Concepts: The UNIX Filesystem

Basic Filesystem Concepts: The UNIX Filesystem

Basic Filesystem Concepts: The UNIX Filesystem

The filesystem is responsible for storing the data on the disk. So to read/write data, it needs to know in which physical blocks the actual data is located; ie how to map files to the disk blocks.

Components of the Berkeley Fast Filesystem:

- set of *inode* storage cells

```
df -i
```

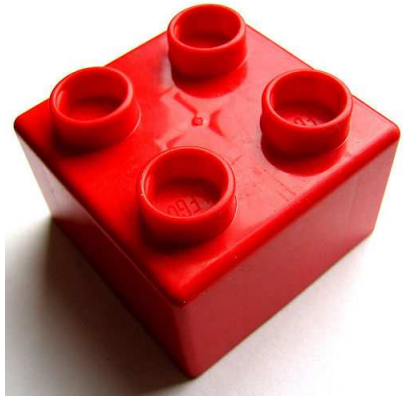
Basic Filesystem Concepts: The UNIX Filesystem

The filesystem is responsible for storing the data on the disk. So to read/write data, it needs to know in which physical blocks the actual data is located; ie how to map files to the disk blocks.

Components of the Berkeley Fast Filesystem:

- set of *inode* storage cells
- set of scattered "superblocks"

```
newfs -N /dev/rdsk/c1t2160d0s0
```



Basic Filesystem Concepts: The UNIX Filesystem

The filesystem is responsible for storing the data on the disk. So to read/write data, it needs to know in which physical blocks the actual data is located; ie how to map files to the disk blocks.

Components of the Berkeley Fast Filesystem:

- set of *inode* storage cells
- set of scattered “superblocks”
- map of disk blocks

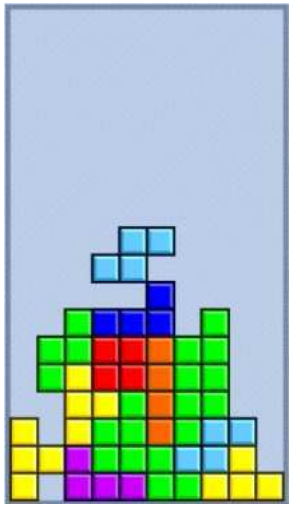
Basic Filesystem Concepts: The UNIX Filesystem

The filesystem is responsible for storing the data on the disk. So to read/write data, it needs to know in which physical blocks the actual data is located; ie how to map files to the disk blocks.

Components of the Berkeley Fast Filesystem:

- set of *inode* storage cells
- set of scattered “superblocks”
- map of disk blocks
- block usage summary

```
fstyp -v /dev/rdsk/c1t2160d0s0
| more
```



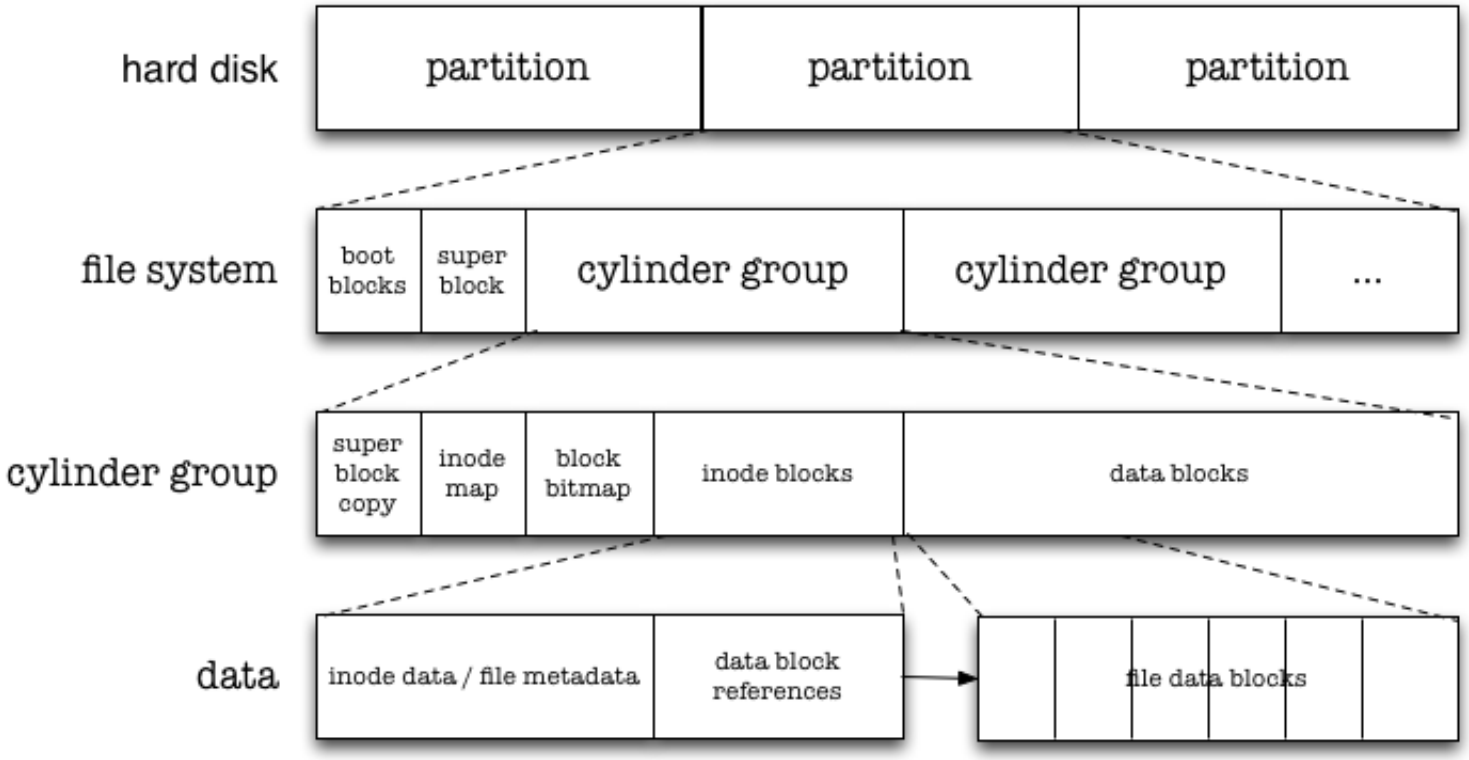
Basic Filesystem Concepts: The UNIX Filesystem

The filesystem is responsible for storing the data on the disk. So to read/write data, it needs to know in which physical blocks the actual data is located; ie how to map files to the disk blocks.

Components of the Berkeley Fast Filesystem:

- set of *inode* storage cells
- set of scattered “superblocks”
- map of disk blocks
- block usage summary
- set of data blocks

Basic Filesystem Concepts: The UNIX Filesystem



Basic Filesystem Concepts: The UNIX Filesystem

Information stored in an *inode*:

Basic Filesystem Concepts: The UNIX Filesystem

Information stored in an *inode*:

- user owner and group owner ID's

Basic Filesystem Concepts: The UNIX Filesystem

Information stored in an *inode*:

- user owner and group owner ID's
- file type

Basic Filesystem Concepts: The UNIX Filesystem

Information stored in an *inode*:

- user owner and group owner ID's
- file type
- access mode (permissions)

Basic Filesystem Concepts: The UNIX Filesystem

Information stored in an *inode*:

- user owner and group owner ID's
- file type
- access mode (permissions)
- file access and modification time

Basic Filesystem Concepts: The UNIX Filesystem

Information stored in an *inode*:

- user owner and group owner ID's
- file type
- access mode (permissions)
- file access and modification time
- file status modification time

Basic Filesystem Concepts: The UNIX Filesystem

Information stored in an *inode*:

- user owner and group owner ID's
- file type
- access mode (permissions)
- file access and modification time
- file status modification time
- number of links to the file

Basic Filesystem Concepts: The UNIX Filesystem

Information stored in an *inode*:

- user owner and group owner ID's
- file type
- access mode (permissions)
- file access and modification time
- file status modification time
- number of links to the file
- size of the file

Basic Filesystem Concepts: The UNIX Filesystem

Information stored in an *inode*:

- user owner and group owner ID's
- file type
- access mode (permissions)
- file access and modification time
- file status modification time
- number of links to the file
- size of the file
- disk device containing this file

Basic Filesystem Concepts: The UNIX Filesystem

Information stored in an *inode*:

- user owner and group owner ID's
- file type
- access mode (permissions)
- file access and modification time
- file status modification time
- number of links to the file
- size of the file
- disk device containing this file

```
$ stat /etc/passwd
```

Basic Filesystem Concepts: The UNIX Filesystem

File types:

- regular files

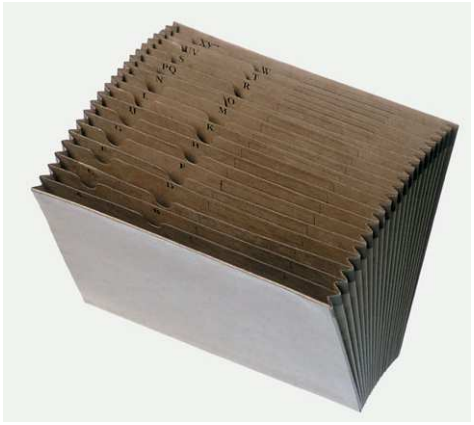
```
$ stat /etc/passwd
```

Basic Filesystem Concepts: The UNIX Filesystem

File types:

- regular files
- directories

```
$ stat /
```



Basic Filesystem Concepts: The UNIX Filesystem

File types:

- regular files
- directories
- special files

```
$ file /dev/* | more
```

Basic Filesystem Concepts: The UNIX Filesystem

File types:

- regular files
- directories
- special files
- links

```
$ touch /tmp/foo
$ ln /tmp/foo /tmp/bar
$ stat /tmp/foo /tmp/bar
$ ln -sf /tmp/foo /tmp/bar
$ stat /tmp/foo /tmp/bar
```

Basic Filesystem Concepts: The UNIX Filesystem

File types:

- regular files
- directories
- special files
- links
- sockets

```
$ stat /dev/log
```

Basic Filesystem Concepts: The UNIX Filesystem

File types:

- regular files
- directories
- special files
- links
- sockets
- named pipes

```
$ mkfifo /tmp/fifo
$ cat /tmp/fifo > /tmp/out &
$ stat /tmp/fifo | tee /tmp/fifo
$ cat /tmp/out
```

Homework

Repeat the examples from class. Make sure you understand the commands and how they relate to the concepts we discussed. Repeat for a different OS, for example:

- `ami-3b361952` – Fedora 23
- `ami-f709a29c` – FreeBSD 10.2
- `ami-569ed93c` – NetBSD 7.0
- `ami-9fbbfaf5` – OmniOS 5.11

Remember to *shut down* your EC2 instances and to *delete* any unused ESB volumes!

`https://www.cs.stevens.edu/~jschauma/615/s16-hw2.html`

Reading

- http://is.gd/5mndwA
- http://is.gd/ig4QP5
- http://is.gd/9YeIKh

Reading

Disk Interfaces:

- SCSI:
  - `http://en.wikipedia.org/wiki/Scsi`
  - scsi(4), scsictl(8);
- ATA:
  - `http://www.ata-atapi.com/`
  - `https://en.wikipedia.org/wiki/Advanced_Technology_Attachment`
  - `https://en.wikipedia.org/wiki/Sata`

Reading

Disk Interfaces:

- Serial attached SCSI:
  - https://en.wikipedia.org/wiki/Serial_attached_SCSI
- Fibre Channel:
  - https://hsi.web.cern.ch/HSI/fcs/fcs.html
  - https://en.wikipedia.org/wiki/Fibrechannel
- AoE, FCoE, iSCSI:
  - https://en.wikipedia.org/wiki/ATA_over_Ethernet
  - https://en.wikipedia.org/wiki/FCoE
  - https://en.wikipedia.org/wiki/ISCSI

Reading

Basic Disk Concepts:

- `http://is.gd/3SXn5N`
- disklabel(8), fdisk(8)
- format(1m)

RAID:

- `https://en.wikipedia.org/wiki/RAID`

Basic Filesystem Concepts:

- `http://is.gd/8KHnQj`
- `http://is.gd/YMm3JL`
- newfs(8)

NFS: `http://is.gd/70yqMZ`