

**VELOCITY**  
**S O F T W A R E**

## *Filesystem Sharing and Cloning with zPRO*

**Velocity Software Inc.**  
196-D Castro Street  
Mountain View CA 94041  
650-964-8867

**Velocity Software GmbH**  
Max-Joseph-Str. 5  
D-68167 Mannheim  
Germany  
+49 (0)621 373844

**Rick Troth**  
**Velocity Software**  
<[rickt@velocitysoftware.com](mailto:rickt@velocitysoftware.com)>  
<http://www.velocitysoftware.com/>  
VM and Linux Workshop 2013  
IUPUI

Copyright © 2013 Velocity Software, Inc. All Rights Reserved.  
Other products and company names mentioned herein may be  
trademarks of their respective owners.

The content of this presentation is informational only and is not intended to be an endorsement by Velocity Software. (ie: I am speaking only for myself.) The reader or attendee is responsible for his/her own use of the concepts and examples presented herein.

In other words: Your mileage may vary. “It Depends.”  
Results not typical. Actual mileage will probably be less.  
Use only as directed. Do not fold, spindle, or mutilate. Not to be taken on an empty stomach. Refrigerate after opening.

In all cases, *“If you can't measure it, I'm just not interested.”*

Some history of shared content

Some ways of sharing content

Some reasons for sharing content

Some solutions to sharing content

Working example with zPRO

# *History of Shared Digital Data*

Tapes

Disks

Network

social/consumer

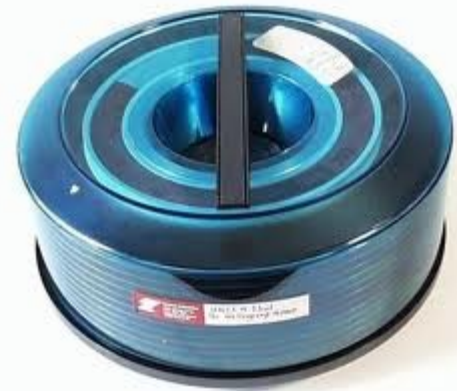
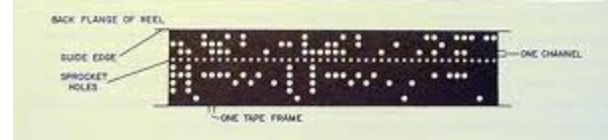
excessive duplication

Only wimps use tape backup: real men just upload their important stuff on ftp, and let the rest of the world mirror it

-- Linus

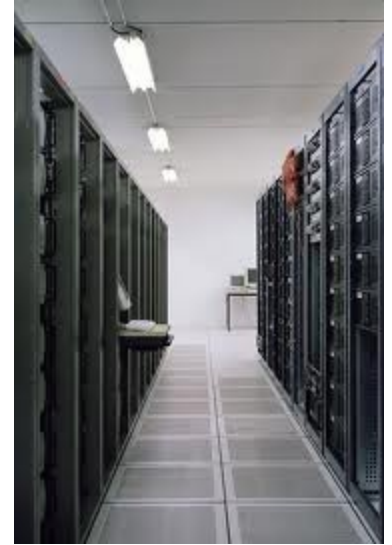
# Data Sharing Methods

Tape, Cards  
Packs, Floppies  
Network Filesystems  
CD ROM, Flash  
Scan Codes  
Network Synchron



# What does “sharing data” mean?

Input/Output  
Immediacy  
Reliability  
Viability  
Security



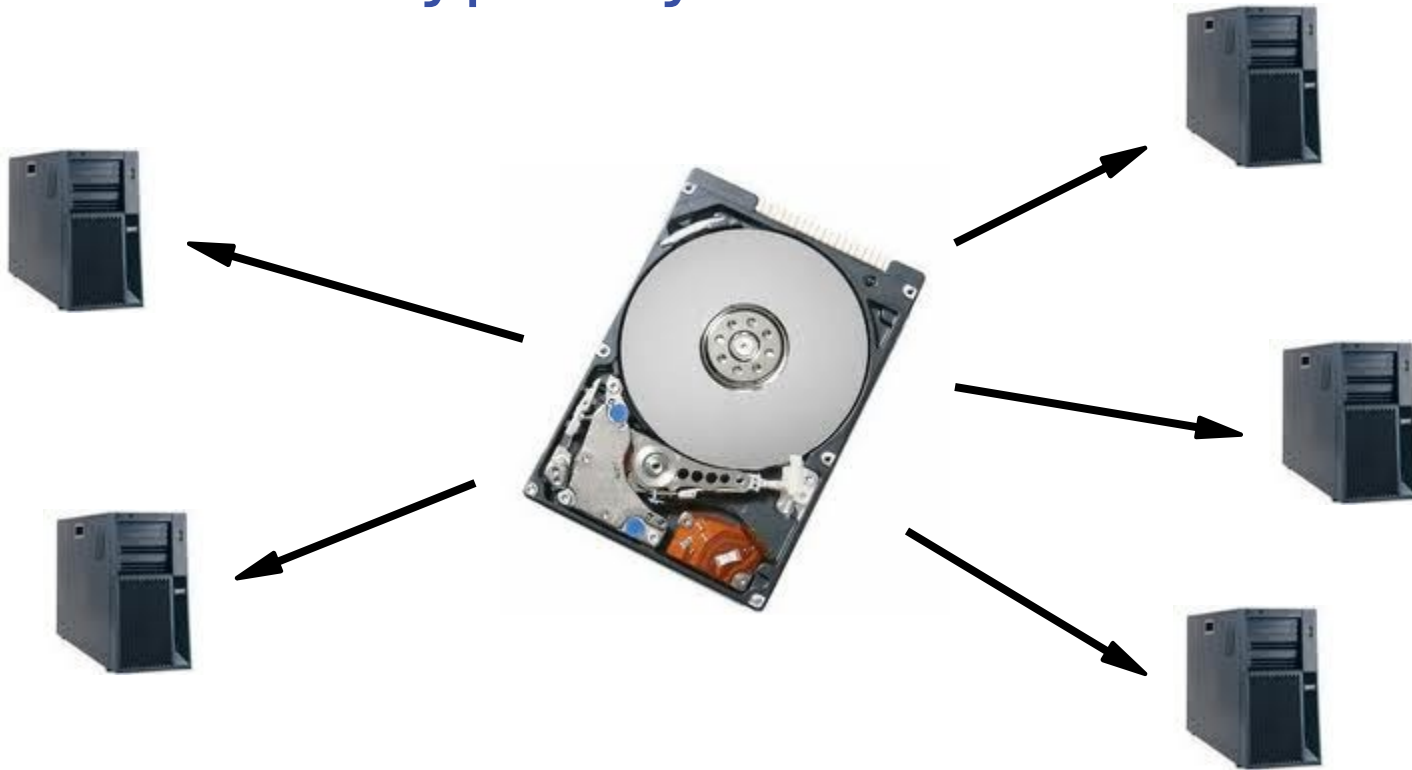
Online -vs- Offline / Dynamic -vs- Resting

# Filesystem Sharing Rationale

Distribution  
Collaboration  
Recovery  
Control  
Deduplication  
Scalability

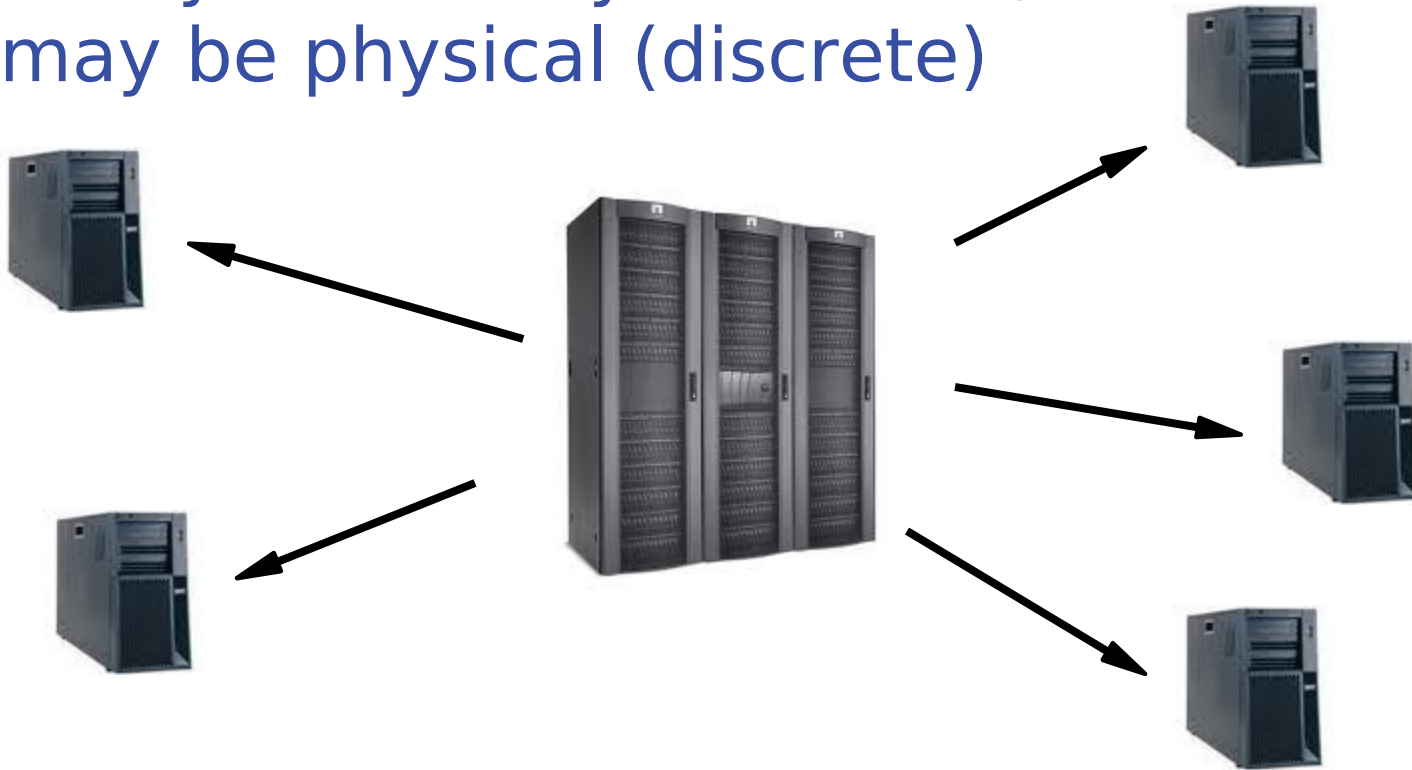


“clients” are typically virtual

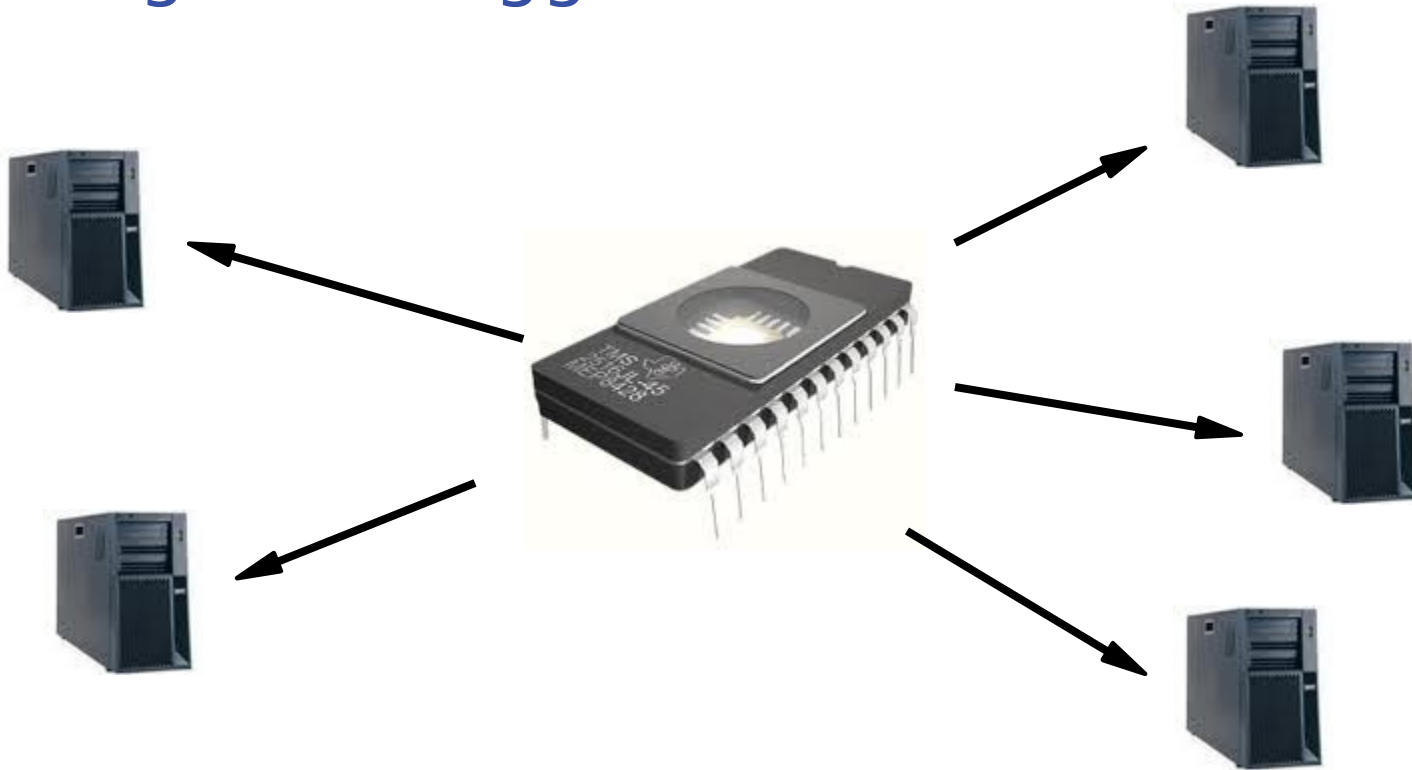




Client systems may be virtual,  
or may be physical (discrete)



Sharing ROM suggests virtual



Standard for z/VM (minidisks)

Must be R/O (block cache)

Candidate FS:

- EXT2 (no journal)
- ISO-9660 (CD-ROM)

VFAT tends to want partitioning

GFS, OCFS2

Shared SAN too (works for physical)

# Shared Disk

```
# df
Filesystem          1K-blocks      Used Available Use% Mounted on
/dev/dasda          476104      394940     56588  88% /Linux-s390
/dev/dasdb          126960       26544     93864  23% /
/dev/dasda          476104      394940     56588  88% /lib
/dev/dasda          476104      394940     56588  88% /bin
/dev/dasda          476104      394940     56588  88% /sbin
/dev/dasda          476104      394940     56588  88% /usr
udev                30580         0      30580   0% /dev
/dev/dasdk          253920     112932    127884  47% /opt/CD2
/dev/dasdm          476104     302828    148700  68% /usr/src
tmpfs               30580         0      30580   0% /tmp
```

R/O media is immutable

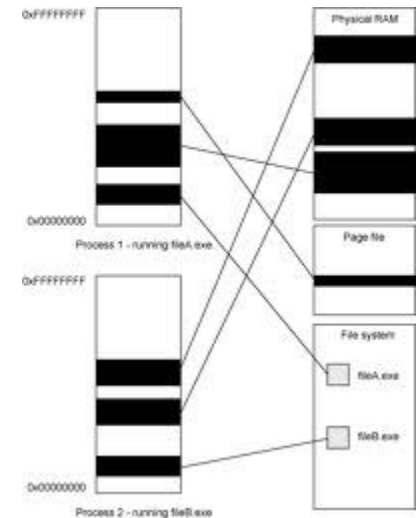
Shared media may be R/O

... ergo ... shared *may* be immutable

Shared memory is common  
DCSS – variable modes

- Restricted – maybe
- TYPE SR

Big boost for CMS  
“back in the day”



# Shared Memory

```
# df
Filesystem          1K-blocks      Used Available Use% Mounted on
/dev/dcssblk0       380888         321900     39328   90% /Linux-s390
/dev/dasdb          126960          24652    95756   21% /
/dev/dcssblk0       380888         321900     39328   90% /lib
/dev/dcssblk0       380888         321900     39328   90% /bin
/dev/dcssblk0       380888         321900     39328   90% /sbin
/dev/dcssblk0       380888         321900     39328   90% /usr
udev                22448            0     22448    0% /dev
/dev/dasdk          253920         112932    127884   47% /opt/CD2
/dev/dasdm          476104         302828    148700   68% /usr/src
tmpfs               22448            0     22448    0% /tmp
```

## The “extreme sport” ... execute-in-place

- No copying of content (disk to memory)
- No I/O
- Just point to it and go!

But ... “binaries are small,  
thus the savings are mediocre at best.”



# Filesystem Sharing History

CMS sharing 190, 19E, others

Solaris sharing of /usr

academic work (AIX/370 and UTS)

Linux/390 and shared /usr

Linux/390 at NW and shared root

RW root with shared op sys

- bind mount selected directories

# Filesystem Sharing with Linux

Shared /usr and others

R/O root with R/W /etc

R/O op sys with R/W root

System maint and package management

Relocatable Packages

DASD on Demand - Disk Automounter

# *Shared op sys -vs- Shared root*

## Install Once, Run Many

- (isn't that why they blessed us with Java?)

## Sharing /usr, /opt, and others,

- so why not also share the root?

## Sharing /bin, /lib, and standard op sys

- works and may be more appealing

Solaris/SunOS supports NFS root  
including read-only /usr content

“Live CD” Linux uses bulk R/O content

- Knoppix, Ubuntu, Kubuntu, recovery tools

USS supports ROR already (Unix on z/OS)

Not weird, Not even new

Many uses, but not widely understood

# *Stability and Manageability*

R/O media is incorruptible

R/O content is centrally maintained

R/O packages are available on-demand

Better D/R – less per-server replication

R/O zLinux no different from R/O PC Linux

# *How to Build Read-Only OS*

Start with standard installation

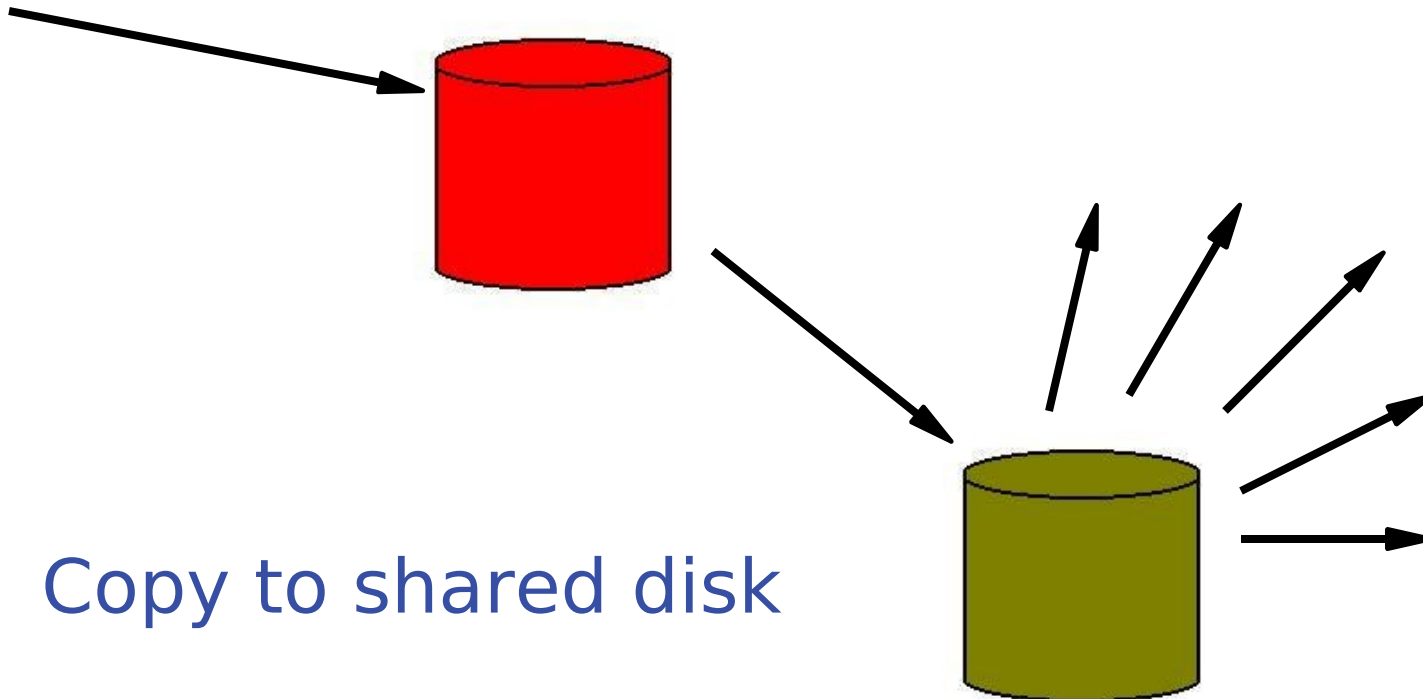
Copy /etc and /var to “run root”

Create other root mount points

Insert /sbin/init+vol script to boot parm

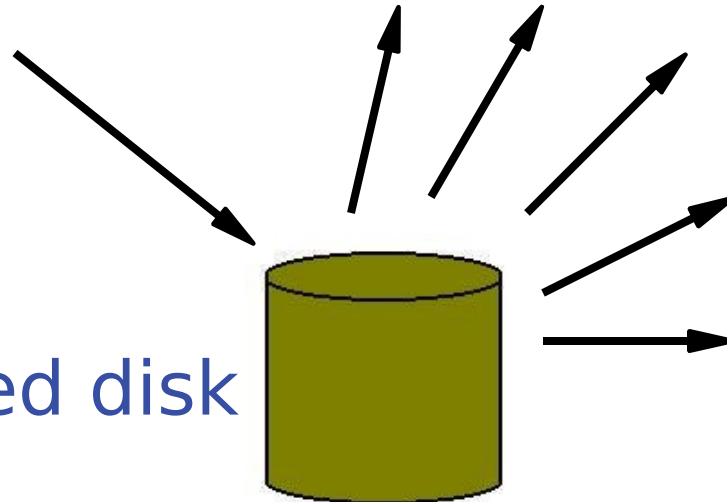
# How to Build Read-Only OS

Start with standard installation



# How to Build Read-Only OS

Do a bunch of prep work ...



... then use shared disk



# */sbin/init+vol Startup Script*

```
#!/bin/sh
mount -r $_RUNFS /mnt
for D in lib bin sbin usr ; do
    mount -o bind /$D /mnt/$D
done
pivot_root /mnt /mnt/$SYSTEM
cd /
exec /sbin/init $*
```

# Reconciling RPM Database

Initial RPM DB matches master

“Client” systems may vary

Master may get updates

... now what?

# Reconciling RPM Database

## Extract master package list

```
# rpm -q -a > master.rpm1
```

## Update client RPM database

```
# for P in `cat master.rpm1`; do  
    rpm -U --justdb $P.rpm ; done
```

# Cloning with zPRO

<http://demo.velocitysoftware.com/zpro/>


System Status	Manage Users	zVM Admin	Velocity Products	Tools	Help
Directory Maintenance					
Cloning					
zPRO Server Expirations					
IP Address Maintenance					
Group Freespace					
zPRO Wizards					




Welcome to zPRO on node VSIVM4



ZPRO Cloud Enablement from Velocity Software

		System Status	Mana
<b>zPRO Serv</b>			
Gold List	Options	Factories	
DEMOLNX			
DEMOZPRO		<b>View</b>	<b>Clone</b>
GOLDLXRO		Work zone	
GOLDR1			
GOLDR2			
GOLDRO			
GOLDVM			



Create a single userid

Choose a “strong” password

Select IP address

Type of minidisk allocation: AUTOG

Location of minidisk allocation: DEMOECKD

Start cloning

Log on ...

- x3270 demo.velocitysoftware.com

Connect to console ...

- ssh demonnnn@demo.velocitysoftware.com

Easy ... and fast!

Demo “GOLDLXRO” uses SuSE R/O root

Combination of ...

- bind-mounted directories, and
- bind-mounted files

# R/O OS with Xen

```
nehemiah:~ # df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/xvdb	5160576	1427492	3523372	29%	/
udev	131168	112	131056	1%	/dev
tmpfs	131168	8	131160	1%	/tmp
/dev/xvdj	20642428	10102248	9491604	52%	/export/home
/dev/xvdk	20642428	176320	19417532	1%	/export/opt
/dev/xvdl	30963708	20238400	9152444	69%	/export/media



# R/O OS with Xen

```
nehemiah:~ # df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/xvda	4127076	1951568	1965864	50%	/Linux-i386
/Linux-i386/lib	4127076	1951568	1965864	50%	/lib
/Linux-i386/bin	4127076	1951568	1965864	50%	/bin
/Linux-i386/sbin	4127076	1951568	1965864	50%	/sbin
/Linux-i386/usr	4127076	1951568	1965864	50%	/usr
/dev/xvdb	5160576	1427500	3523364	29%	/
udev	131168	112	131056	1%	/dev
tmpfs	131168	8	131160	1%	/tmp
/dev/xvdj	20642428	10102248	9491604	52%	/export/home
/dev/xvdk	20642428	176320	19417532	1%	/export/opt
/dev/xvdl	30963708	20238400	9152444	69%	/export/media

# R/O OS with Xen

```
nehemiah:~ # df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/xvda	4127076	1951568	1965864	50%	/Linux-i386
/dev/xvdb	5160576	1427500	3523364	29%	/
udev	131168	112	131056	1%	/dev
tmpfs	131168	8	131160	1%	/tmp
/dev/xvdj	20642428	10102248	9491604	52%	/export/home
/dev/xvdk	20642428	176320	19417532	1%	/export/opt
/dev/xvdl	30963708	20238400	9152444	69%	/export/media

# R/O OS with Xen

```
obadiah:~ # df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/xvda	4127076	1951568	1965864	50%	/Linux-i386
/dev/xvdb	4128448	1927680	1991056	50%	/
udev	32864	104	32760	1%	/dev
tmpfs	32864	16	32848	1%	/tmp

```
disk=[ 'file:/var/vmimage/nehemiah/disk0.xvd,xvda,r',  
       'phy:/dev/sysvg1/nehemiah,xvdb,w',  
       ... ]
```

```
-rw----- 5 root root 4294967296 2011-03-25 09:07  
           /var/vmimage/nehemiah/disk0.xvd
```

Standard for z/VM (host disks or “full pack”)  
Increasingly popular with Linux  
Also mount-by-uuid (works for swap)  
Does not require partitioning  
Consistent across architectures

Use 'rsync'

Could replace all other Unix backup tools

Use 'rsync'

Could replace all other Unix backup tools

Rick's preferred options:

**-a -u -x -H -K -O -S --safe-links**

**rsync** *[options] source/. target/.*

NFS ... and/or SMB

CD-ROM

USB, flash

'vmlink'

DCSS



# Automating Disk Attachment

```
#  
# /etc/auto.master  
#  
/home    /etc/auto.home  
/misc    /etc/auto.misc  
/dasd    /etc/auto.dasd
```

# Automating DCSS Attachment

```
#  
# /etc/auto.master  
#  
/home    /etc/auto.home  
/misc    /etc/auto.misc  
/dasd    /etc/auto.dasd  
/dcss    /etc/auto.dcss
```

Partitioning is another layer,  
added complexity

Partitioning may not be needed,  
find out if it is ... or not

Certain (non-Linux and non-VM)  
systems or environments expect it

# About Partitioning

CDL if you need to share with z/OS

“CMS RESERVE” for direct sharing with CMS

Traditional (PC) partition table  
makes Windows happier



# About Partitioning

## On the host ...

```
# ls -lad *.fba
-rw-rw---- 1 rmt  root 402653184 2011-09-18 19:41 01b0.fba
-rw-rw---- 1 rmt  root  67108864 2012-05-30 14:48 01b1.fba
lrwxrwxrwx 1 root  root           8 2012-02-26 21:00 01bf.fba -> /dev/sda
```

## Easy maint access ...

```
# mount -o loop 01b1.fba /mnt
```

## On the “guest” ...

```
# df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/dasda	516040	322216	167612	66%	/Linux-s390
/dev/dasdb	63472	41532	18664	69%	/

# Relocatable Packages

Deploy instantly

Good candidates for shared FS

- Less content to be backed up

Good candidates for R/O media

- Protected copies (R/O to each client)

Non-intrusive (to the guest op sys)

Non-disruptive (to the users and work)

Mixed releases as needed

Wide spectrum of data sharing options  
File and Filesystem Sharing is rock solid

Consider your needs, familiarize the team,  
make a plan and execute

The real advantage is not storage savings  
but management of myriad systems