Filesystem Sharing and Cloning with zPRO

Rick Troth
Velocity Software
<rickt@velocitysoftware.com>
http://www.velocitysoftware.com/
VM and Linux Workshop 2013
IUPUI
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.”
Filesystem Sharing

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 Synch
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)
Shared FS in ROM

Sharing ROM suggests virtual
Shared Disk

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)
# df

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

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

Copy to shared disk
How to Build Read-Only OS

Do a bunch of prep work ...

... then use shared disk
#!/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?
Extract master package list
# rpm -q -a > master.rpml

Update client RPM database
# for P in `cat master.rpml` ; do
  rpm -U --justdb $P.rpm ; done
Cloning with zPRO

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

Welcome to zPRO on node VSIVM4

zPRO Cloud Enablement from Velocity Software
Create a single userid
Choose a “strong” password
Select IP address
Type of minidisk allocation: AUTOG
Location of minidisk allocation: DEMOECKD
Start cloning
Cloning with zPRO

Log on ...

- x3270 demo.velocitysoftware.com

Connect to console ...

- ssh demonnnnn@demo.velocitysoftware.com

Easy ... and fast!
Demo “GOLDLXRO” uses SuSE R/O root

Combination of …

- bind-mounted directories, and
- bind-mounted files
```bash
nehemiah:~ # df

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>1K-blocks</th>
<th>Used</th>
<th>Available</th>
<th>Use%</th>
<th>Mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/xvdb</td>
<td>5160576</td>
<td>1427492</td>
<td>3523372</td>
<td>29%</td>
<td>/</td>
</tr>
<tr>
<td>udev</td>
<td>131168</td>
<td>112</td>
<td>131056</td>
<td>1%</td>
<td>/dev</td>
</tr>
<tr>
<td>tmpfs</td>
<td>131168</td>
<td>8</td>
<td>131160</td>
<td>1%</td>
<td>/tmp</td>
</tr>
<tr>
<td>/dev/xvdj</td>
<td>20642428</td>
<td>10102248</td>
<td>9491604</td>
<td>52%</td>
<td>/export/home</td>
</tr>
<tr>
<td>/dev/xvdk</td>
<td>20642428</td>
<td>176320</td>
<td>19417532</td>
<td>1%</td>
<td>/export/opt</td>
</tr>
<tr>
<td>/dev/xvdl</td>
<td>30963708</td>
<td>20238400</td>
<td>9152444</td>
<td>69%</td>
<td>/export/media</td>
</tr>
</tbody>
</table>
```
nehemiah:~ # df

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

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>1K-blocks</th>
<th>Used</th>
<th>Available</th>
<th>Use%</th>
<th>Mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/xvda</td>
<td>4127076</td>
<td>1951568</td>
<td>1965864</td>
<td>50%</td>
<td>/Linux-i386</td>
</tr>
<tr>
<td>/dev/xvdb</td>
<td>5160576</td>
<td>1427500</td>
<td>3523364</td>
<td>29%</td>
<td>/</td>
</tr>
<tr>
<td>udev</td>
<td>131168</td>
<td>112</td>
<td>131056</td>
<td>1%</td>
<td>/dev</td>
</tr>
<tr>
<td>tmpfs</td>
<td>131168</td>
<td>8</td>
<td>131160</td>
<td>1%</td>
<td>/tmp</td>
</tr>
<tr>
<td>/dev/xvdj</td>
<td>20642428</td>
<td>10102248</td>
<td>9491604</td>
<td>52%</td>
<td>/export/home</td>
</tr>
<tr>
<td>/dev/xvdk</td>
<td>20642428</td>
<td>176320</td>
<td>19417532</td>
<td>1%</td>
<td>/export/opt</td>
</tr>
<tr>
<td>/dev/xvd1</td>
<td>30963708</td>
<td>20238400</td>
<td>9152444</td>
<td>69%</td>
<td>/export/media</td>
</tr>
</tbody>
</table>
obadiah:~ # df

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>1K-blocks</th>
<th>Used</th>
<th>Available</th>
<th>Use%</th>
<th>Mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/xvda</td>
<td>4127076</td>
<td>1951568</td>
<td>1965864</td>
<td>50%</td>
<td>/Linux-i386</td>
</tr>
<tr>
<td>/dev/xvdb</td>
<td>4128448</td>
<td>1927680</td>
<td>1991056</td>
<td>50%</td>
<td>/</td>
</tr>
<tr>
<td>udev</td>
<td>32864</td>
<td>104</td>
<td>32760</td>
<td>1%</td>
<td>/dev</td>
</tr>
<tr>
<td>tmpfs</td>
<td>32864</td>
<td>16</td>
<td>32848</td>
<td>1%</td>
<td>/tmp</td>
</tr>
</tbody>
</table>
disk=[ 'file:/var/vmachine/nehemiah/disk0.xvd,xvda,r',
    'phy:/dev/sysvg1/nehemiah,xvdb,w',
    ... ]

-rw------- 5 root root 4294967296 2011-03-25 09:07
/var/vmachine/nehemiah/disk0.xvd
Mount by Label

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
About Backups

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

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>1K-blocks</th>
<th>Used</th>
<th>Available</th>
<th>Use%</th>
<th>Mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/dasda</td>
<td>516040</td>
<td>322216</td>
<td>167612</td>
<td>66%</td>
<td>/Linux-s390</td>
</tr>
<tr>
<td>/dev/dasdb</td>
<td>63472</td>
<td>41532</td>
<td>18664</td>
<td>69%</td>
<td>/</td>
</tr>
</tbody>
</table>
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
Summary

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