Sharing and maintaining Linux under z/VM

Michael MacIsaac - mikemac@us.ibm.com
Wednesday January 23rd, 2008

Abstract

Large operating systems leverage shared file structures to get the benefits of reduced disk space, simplified maintenance and simplified systems management. The draft Redpaper, "Sharing and maintaining Linux under z/VM", written in conjuction with Nationwide Insurance, describes how to create a Linux solution with shared file systems under z/VM. It also describes a maintenance system where the same Linux image exists on a test, maintenance and gold virtual servers. It is based on z/VM 5.3 and SLES 10.

Agenda

- Introductions
- Overview
- Thinking about "systems"
- Block diagrams
- The "guts" of the paper
- Live demo!
- Resources
- Questions

Introductions

- Who am I?
  - Mike MacIsaac, mikemac@us.ibm.com
  - 20+ years at IBM in NY
    - Programmer
    - OS/390, USS, Redbook project lead
    - Marketing technical support of z/VM, Linux, IBM software, ...
    - New: z/VM development manager (check technical skills at the door :))
- Who are you?
  - Time spent administering z/VM and Linux
    - z/VM, almost no Linux
    - Mostly z/VM, some Linux
    - Mostly Linux, some z/VM
    - Other
  - IT status:
    - Do you have Linux and z/VM in production?
    - In test?
    - Planning a proof of concept?
    - Other
- Something you are hoping to get out of this talk?

Overview:

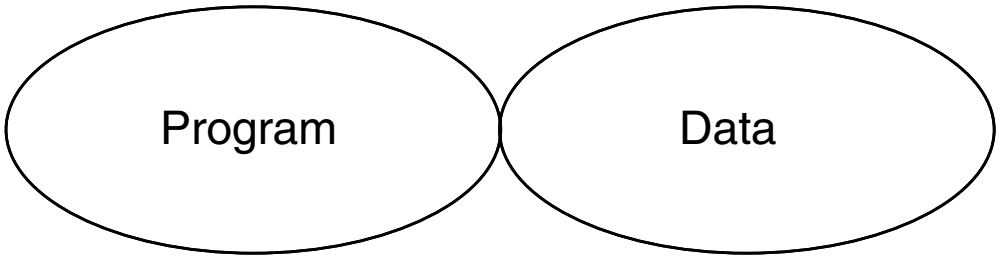
- History:
  - Paper on "Read-only root" started by Steve Womer (writing) and Rick Troth (technical)
  - Nationwide approached the ITSO
    - March - July 2007: logistics/political battles - tide's in, tide's out
  - Was written mostly in August 2007
  - Presented at SHARE by Rick Troth in August 2007
- Philosophy (from Rick Troth's SHARE presentation)
  - Document how to share the root file system (moniker: "Read-only root")
  - Install Once, Run Many
  - Sharing /usr, /opt, and others so why not also share the root?
- This presentation is a shameful plug for the Redpaper :))

Agenda

- ~~Introductions~~
- ~~Overview~~
- Thinking about "systems"
- Block diagrams
- The "guts" of the paper
- Live demo!
- Resources
- Questions

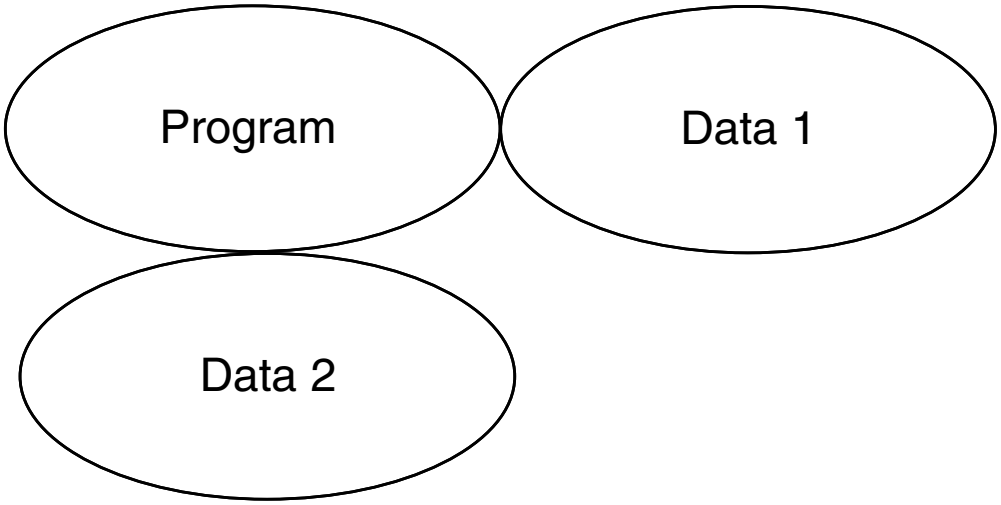
Thinking about systems

- Computing model in the dark ages - no OS needed



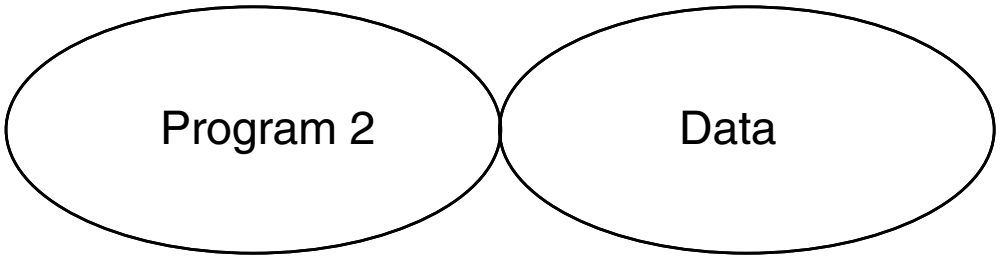
Thinking about systems (cont'd)

- Processing another data set



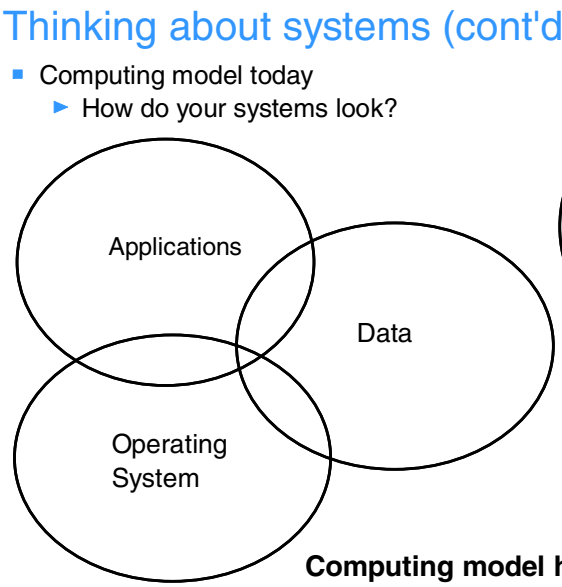
Thinking about systems (cont'd)

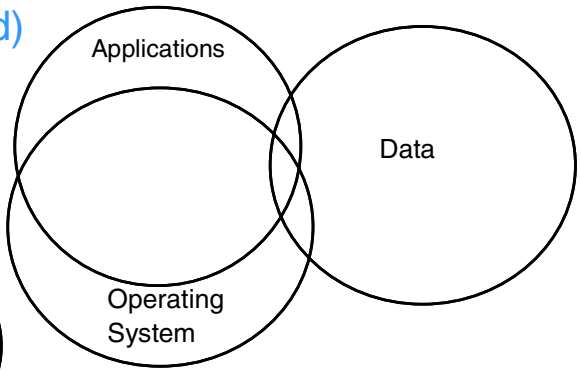
- Running a different program
- It was very easy to keep programs and data separate

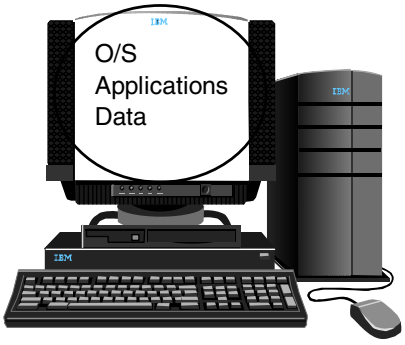


Thinking about systems (cont'd)

- Computing model today
  - How do your systems look?



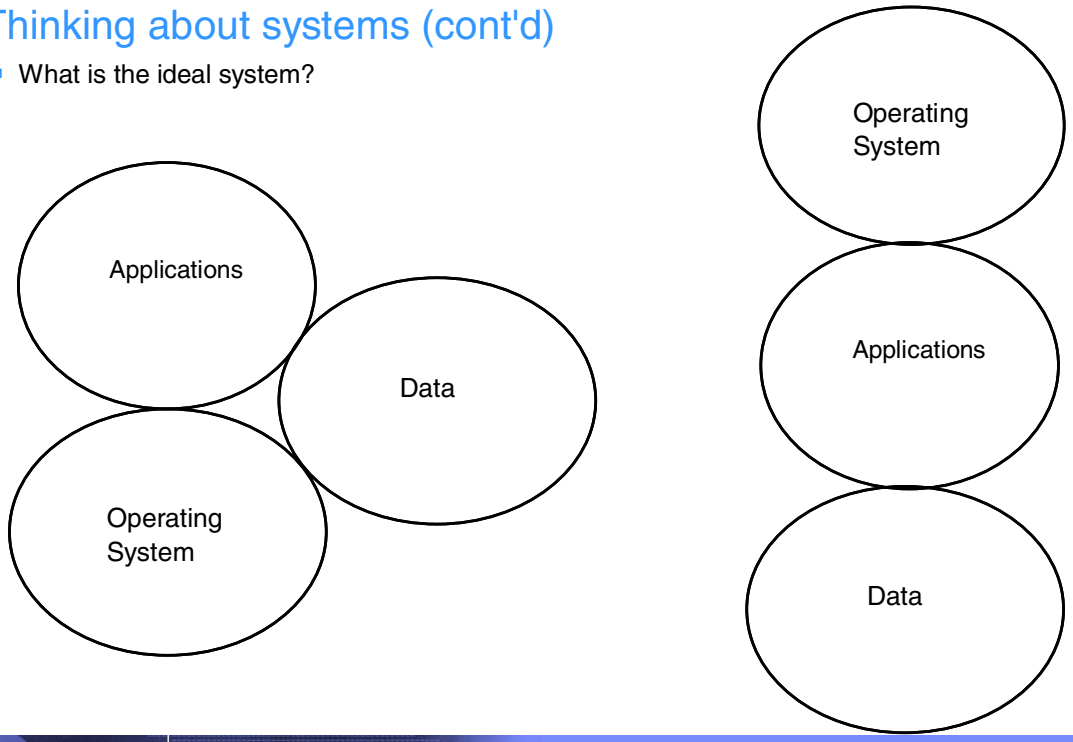




Computing model hierarchy:
Hardware
Operating System
Applicatons
Data

Thinking about systems (cont'd)

- What is the ideal system?



FHS* summary

Directory	Description
/	Root file system (must be able to boot/repair)
/bin	Essential commands (static)
/boot	Static files of the boot loader (static)
/dev	Device files (static, maintained by OS)
/etc	Host-specific system configuration (static)
/etc/opt	Add-on application configuration (site specific)
/home	User's home directories (site-specific, optional)
/lib	Essential shared libraries and kernel modules
/media	Mount point for removeable media (N/A on System z)
/mnt	Temporary mount point (usually empty)
/opt	Add-on application software packages (site-specific)
/root	Root user home directory (static, optional)
/sbin	Essential system binaries (static)
/srv	Data for services provided by this system (site specific)
/tmp	Temporary files (delete when system is booted?)
/usr	Sharable read-only data
/usr/bin	Most user commands
/usr/include	Header files included by C programs
/usr/lib	Libraries
/usr/local	Local hierarchy (empty after main installation)
/usr/sbin	Non-vital system binaries
/usr/share	Architecture-independent data
/var	Variable data
/var/cache	Application cache data
/var/lib	Variable state information
/var/local	Variable data for /usr/local
/var/lock	Lock files
/var/log	Log files and directories
/var/opt	Variable data for /opt
/var/run	Data relevant to running processes
/var/spool	Application spool data
/var/tmp	Temporary files preserved between system reboots

*Filesystem Hierarchy Standard - reference: http://www.pathname.com/fhs/pub/fhs-2.3.pdf

FHS (cont'd)

Directory	Description
/	Root file system (must be able to boot/repair)
/bin	Essential commands (static)
/boot	Static files of the boot loader (static)
/dev	Device files (static, maintained by OS)
/etc	Host-specific system configuration (static)
/etc/opt	Add-on application configuration (site specific)
/home	User's home directories (site-specific, optional)
/lib	Essential shared libraries and kernel modules
/media	Mount point for removeable media (N/A on System z)
/mnt	Temporary mount point (usually empty)
/opt	Add-on application software packages (site-specific)
/root	Root user home directory (static, optional)
/sbin	Essential system binaries (static)
/srv	Data for services provided by this system (site specific)
/tmp	Temporary files (delete when system is booted?)
/usr	Sharable read-only data
/usr/bin	Most user commands
/usr/include	Header files included by C programs
/usr/lib	Libraries
/usr/local	Local hierarchy (empty after main installation)
/usr/sbin	Non-vital system binaries
/usr/share	Architecture-independent data
/var	Variable data
/var/cache	Application cache data
/var/lib	Variable state information
/var/local	Variable data for /usr/local
/var/lock	Lock files
/var/log	Log files and directories
/var/opt	Variable data for /opt
/var/run	Data relevant to running processes
/var/spool	Application spool data
/var/tmp	Temporary files preserved between system reboots

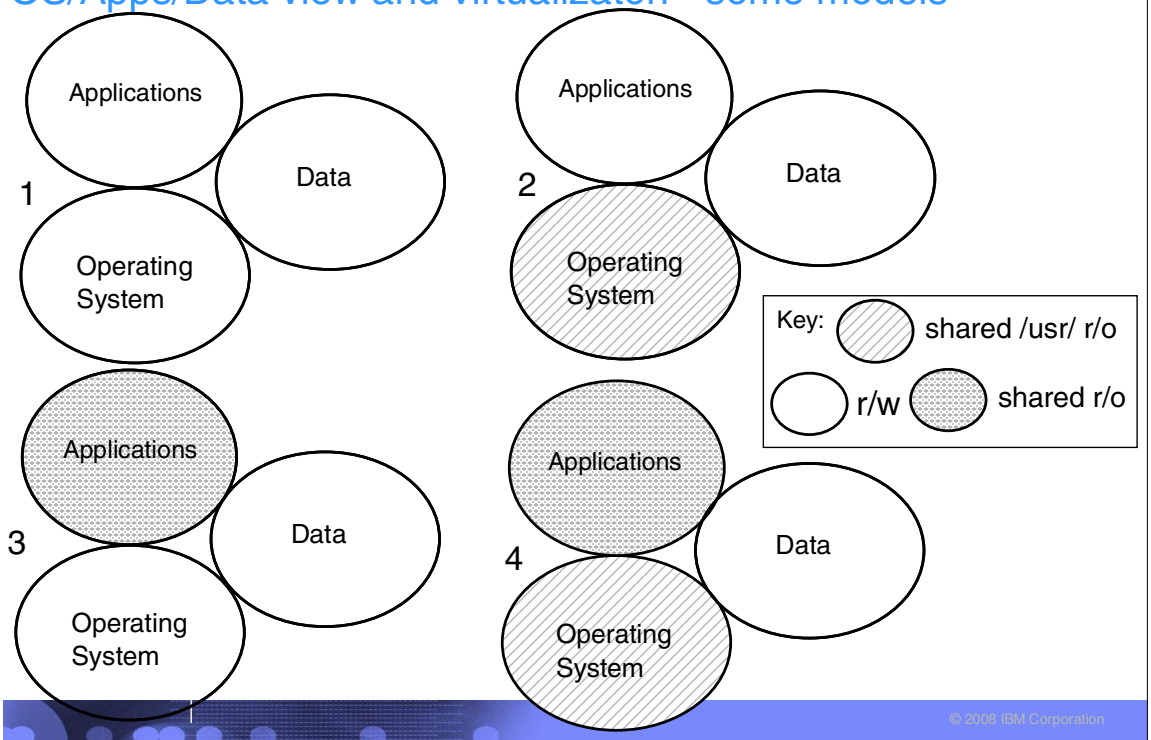
Data → /home

Apps → /opt

Data → /srv

Data or OS? → /var

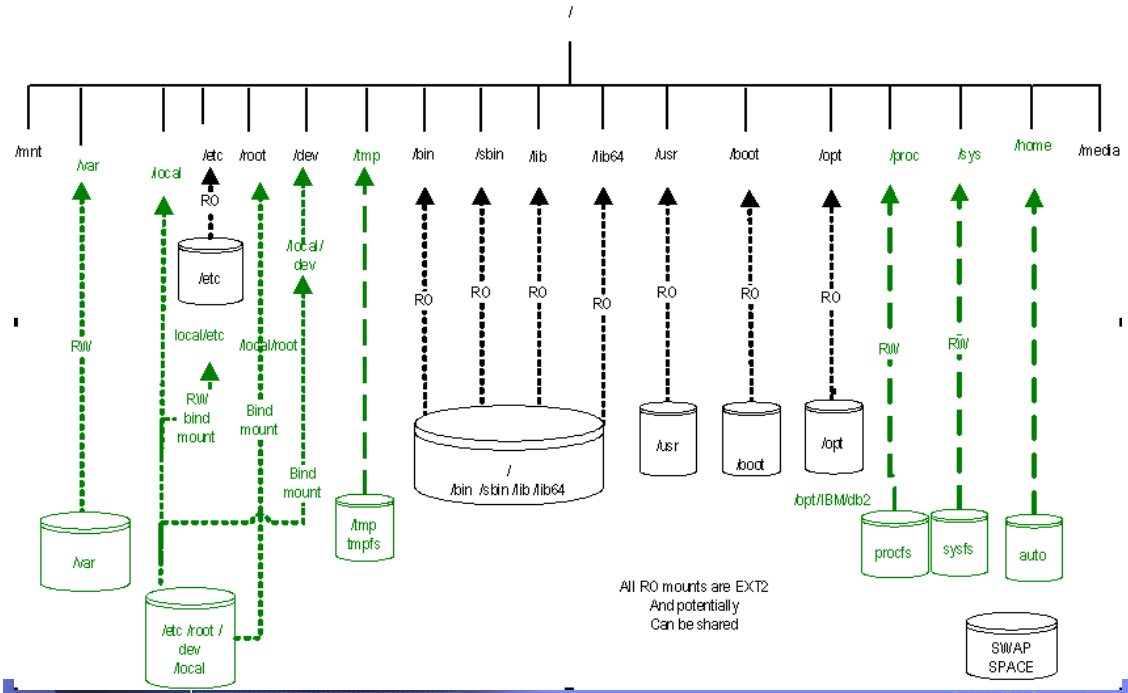
OS/Apps/Data view and virtualizaton - some models



Agenda

- ~~Introductions~~
- ~~Overview~~
- ~~Thinking about "systems"~~
- Block diagrams
- The "guts" of the paper
- Live demo!
- Resources
- Questions

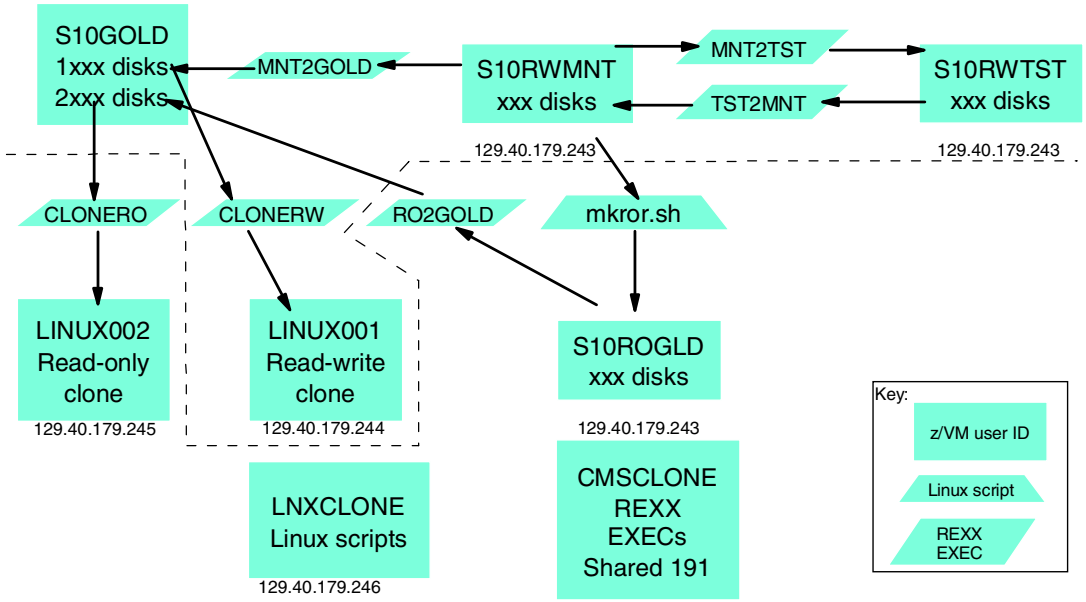
File systems block diagram



File systems - table view

Directory	FS type	Attributes	Device	Vaddr
/	ext2	R/O	/dev/dasdb1	1B1
/bin/	ext2	R/O		
/boot/	ext2	R/O	/dev/dasda1	1B0
/dev/	udev	R/W		
/etc/	bind mount	R/W		
/home/	auto mount	R/W		
/lib/, /lib64/	ext2	R/O		
/local	ext3	R/W	/dev/dasdf1	1B5
/media/	ext2	R/O		
/mnt/	ext2	R/O		
/opt/	ext2	R/O	/dev/dasdi1	1B8
/proc/	procfs	R/W		
/root/	bind	R/W		
/sbin/	ext2	R/O		
/sys/	sysfs	R/W		
/tmp/	tmpfs	R/W		
/usr/	ext2	R/O	/dev/dasdh1	1B7
/var/	ext3	R/W	/dev/dasdg1	1B6
/srv/	ext3	R/W		
swap 1	swap	R/W	/dev/dasdc1	1B2
swap 2	swap	R/W	/dev/dasdd1	1B3
swap 3	swap	R/W	/dev/dasde1	1B4

System block diagram



Agenda

- ~~Introductions~~
- ~~Overview~~
- ~~Thinking about "systems"~~
- ~~Block diagrams~~
- The "guts" of the paper
- Live demo!
- Resources
- Questions

The guts (adapted from Rick Troth's SHARE pitch)

- "Share more stuff"
  - Install Once, Run Many - an old Gospel, fully realized
  - Sharing /usr, /opt, and others so why not also share the root?
- Sharing options
  - NFS
  - SMB (SAMBA)
  - VM minidisk <= today
  - SAN <= future
- How to build a read-only root system
  - Start with “monolithic” distro installation, do "minor prep"
  - Copy to eventual R/O
  - Create reference /local
  - Replace boot.rootfsck
    - Does not check root (1b1 disk)
    - Checks and mounts /local (1b5 disk)
    - Bind mounts /etc, /dev, and /root
    - Happens during “boot” run level
- Summary
  - The real advantage is not space savings but is management of myriad systems
  - Start with one read-only package or directory or disk and grow from there

The guts (from the Redpaper)

- The Redpaper diverged from the Nationwide system
  - HOWTO vs. Production
- How to build a **read-write** system
  - Create the first z/VM user IDs
  - Populate the CMS minidisks on CMSCLONE
  - Install SLES 10 onto S10RWMNT
  - Install and customize SLES 10 onto LNXCLONE
  - Copy Linux from S10RWMNT to S10RWTST
  - Customize and test Linux on S10RWTST
  - Copy Linux back from S10RWTST to S10RWMNT
  - Copy Linux from S10RWMNT to gold disks
  - Clone a read-write Linux to S10RWTST
- How to build a **read-only root** system
  - Define a user ID, S10ROGLD, for the first read-only root system
  - Create a prototype system and test
  - Copy read-only Linux to gold disks
  - Clone a read-only Linux

The guts of the Redpaper (cont'd)

- Creating a user ID for a common CMS disk

```
USER CMSCLONE PASSWD 256M 1G BG
 INCLUDE IBMDFLT
 OPTION APPLMON LNKNOPAS
 IPL CMS
 MACHINE ESA 4
 MDISK 0191 3390 0001 0030 MM912F MR PASSWD PASSWD PASSWD
 MDISK 0192 3390 0031 0100 MM912F MR ALL PASSWD PASSWD
```

- Creating a PROFILE for Linux

```
PROFILE RORDFLT
  IPL CMS
  MACHINE ESA 4
  CPU 00 BASE
  CPU 01
  NICDEF 600 TYPE QDIO LAN SYSTEM VSW1
  SPOOL 000C 2540 READER *
  SPOOL 000D 2540 PUNCH A
  SPOOL 000E 1403 A
  CONSOLE 009 3215 T
  LINK MAINT 0190 0190 RR
  LINK MAINT 019D 019D RR
  LINK MAINT 019E 019E RR
  LINK CMSCLONE 192 191 RR
  LINK TCPMAINT 592 592 RR
```

The guts of the Redpaper (cont'd)

- Creating a user ID for first Linux system

```
USER S10RWMNT PASSWD 256M 1G G
 INCLUDE RORDFLT
 OPTION APPLMON
 MDISK 01B0 3390 0001 0030 MM9131 MR PASSWD PASSWD PASSWD
 MDISK 01B1 3390 0031 0400 MM9131 MR PASSWD PASSWD PASSWD
 MDISK 01B4 3390 0431 0550 MM9131 MR PASSWD PASSWD PASSWD
 MDISK 01B5 3390 0981 0100 MM9131 MR PASSWD PASSWD PASSWD
 MDISK 01B6 3390 1081 0400 MM9131 MR PASSWD PASSWD PASSWD
 MDISK 01B7 3390 1481 1750 MM9131 MR PASSWD PASSWD PASSWD
 MDISK 01B8 3390 3231 0108 MM9131 MR PASSWD PASSWD PASSWD
```

- Creating a NOLOG user ID for storing systems

```
USER S10GOLD NOLOG 256M 1G G
 INCLUDE RORDFLT
 OPTION APPLMON
 MDISK 11B0 3390 0001 0030 MM912E MR PASSWD PASSWD PASSWD
 MDISK 11B1 3390 0031 0400 MM912E MR PASSWD PASSWD PASSWD
 MDISK 11B4 3390 0431 0550 MM912E MR PASSWD PASSWD PASSWD
 MDISK 11B5 3390 0981 0100 MM912E MR PASSWD PASSWD PASSWD
 MDISK 11B6 3390 1081 0400 MM912E MR PASSWD PASSWD PASSWD
 MDISK 11B7 3390 1481 1750 MM912E MR PASSWD PASSWD PASSWD
 MDISK 11B8 3390 3231 0108 MM912E MR PASSWD PASSWD PASSWD
 MDISK 21B0 3390 0001 0030 MM9130 MR PASSWD PASSWD PASSWD
 MDISK 21B1 3390 0031 0400 MM9130 MR PASSWD PASSWD PASSWD
 MDISK 21B4 3390 0431 0550 MM9130 MR PASSWD PASSWD PASSWD
 ...
```


The guts of the Redpaper (cont'd)

- Populating the CMSCLONE 192 disk with the files:
  - S10RWTST PARM-S10 Parameter file for the user ID S10RWTST.
  - SLES10 EXEC EXEC to invoke the SLES 10 installation.
  - SWAPGEN EXEC EXEC to create VDISK swap spaces.
  - PROFILE EXEC EXEC IPL Linux from 1B0 on a disconnected machine
  - PROFILE XEDIT Configuration file that is read when XEDIT is invoked.
  - SLES10 KERNEL The SLES 10 kernel.
  - SLES10 INITRD The SLES 10 initial RAMdisk.
  - <userid> PARM-S10 Parameter files for other user IDs
- Installing SLES 10 onto S10RWTST - minimal system
  - Follow minidisk layout in previous table
- Installing and customizing SLES 10 onto LNXCLONE
  - Single 3390-3 minidisk for root fs at 1B0
  - Main function is to create a read-only root system, via script **mkror.sh**
  - Copy tar file associated with Redpaper
  - vmcp module is set to load at boot time for CP commands
- Copying Linux from S10RWMNT to S10RWTST

```
==> mnt2tst
Do you want to copy R/W disks from S10RWMNT to S10RWTST? y/n
y
Copying minidisk 01B0 to 11B0 ...
00: Command complete: FLASHCOPY 01B0 0 END TO 11B0 0 END
...
```

The guts - TST2MNT EXEC (updated by Jim Vincent)

```
...
/* Copy Linux from maintenance ID to test ID */
Address 'COMMAND'
source = 'S10RWTST'
target = 'S10RWMNT'
Say 'Do you want to copy R/W disks from' source 'to' target'? y/n'
Parse Upper Pull answer .
If (answer \= 'Y') Then Exit 1
ro_addrs = '01B0 01B1 01B4 01B5 01B6 01B7 01B8'
rw_addrs = '11B0 11B1 11B4 11B5 11B6 11B7 11B8'
Do a = 1 to Words(rw_addrs)  /* link target disks R/W */
  addr = Word(rw_addrs,a)
  'CP LINK' target Right(addr,3) addr 'MR'
  If (rc \= 0) Then Call CleanUp 100+a
  End
Do a = 1 to Words(ro_addrs)  /* link source disks R/O */
  addr = Word(ro_addrs,a)
  'CP LINK' source addr addr 'RR'
  If (rc \= 0) Then Call CleanUp 200+a
  End
Do a = 1 to Words(ro_addrs)  /* copy disks */
  source_addr = Word(ro_addrs,a)
  target_addr = Word(rw_addrs,a)
  'EXEC COPYMDSK' source_addr target_addr
  If (rc \= 0) Then Call CleanUp 300+a
  End
...
```


The guts - COPYMDSK EXEC (updated by Jim Vincent)

```
/* COPYMDSK EXEC - copy minidisk w/FLASHCOPY, if it fails, try DDR   */
/*  Parm 1: vaddr of source minidisk                                 */
/*  Parm 2: vaddr of target minidisk                                 */
Address 'COMMAND'
Parse Arg source target .
Say
Say 'Copying minidisk' source 'to' target '...'
'CP FLASHCOPY' source '0 END' target '0 END'
If (rc \= 0) Then Do                  /* Fallback to DDR             */
  Say 'FLASHCOPY failed, falling back to DDR ...'
                                      /* Queue up DDR commands       */
  Queue 'SYSPRINT CONS'               /* Don't print to file         */
  Queue 'PROMPTS OFF'                 /* Don't ask 'Are you sure?'   */
  Queue 'IN' source '3390'            /* Input minidisk              */
  Queue 'OUT' target '3390'           /* Output minidisk             */
  Queue 'COPY ALL'                    /* Copy all contents           */
  Queue ' '                           /* Empty record ends DDR       */
  'DDR'
  retVal = rc
  End
Else retVal = rc
Say 'Return value =' retVal
Return retVal
```

The guts of the Redpaper (cont'd)

- Recommended configuration changes for "gold" Linux
  - Parameter line and menu delay changed in /etc/zipl.conf
  - “timer pop” is turned off
  - CMS file system (cmsfs) package installed
  - cmm module set to load at boot time
  - System set to halt, not reboot when trapping Ctrl-Alt-Del
  - boot.findself script copied for network self-discovery
  - Mount points created under /opt/ for mounting middleware
  - cloneprep.sh copied to clean up before cloning

- Copying Linux from S10RWTST to S10RWMNT

```
==> tst2mnt
Do you want to copy R/W disks from S10RWTST to S10RWMNT? y/n
y
Copying minidisk 01B0 to 11B0 ...
...
```


The guts of the Redpaper (cont'd)

- Copying Linux from S10RWMNT to "gold disks"

```
==> mnt2gold
Do you want to copy R/W disks from S10RWMNT to S10GOLD? y/n
y

Copying minidisk 01B0 to 11B0 ...
00: Command complete: FLASHCOPY 01B0 0 END TO 11B0 0 END
Return value = 0

Copying minidisk 01B1 to 11B1 ...
00: HCPCMM296E Status is not as required - 01B1; an unexpected condition
00: HCPCMM296E occurred while executing a FLASHCOPY command, code = A7.
FLASHCOPY failed, falling back to DDR ...
z/VM DASD DUMP/RESTORE PROGRAM
HCPDDR696I VOLID READ IS 0X01B1
HCPDDR696I VOLID READ IS 0X01B1
COPYING   0X01B1
COPYING DATA  10/12/07 AT 16.47.31  GMT FROM 0X01B1 TO 0X01B1
INPUT CYLINDER EXTENTS      OUTPUT CYLINDER EXTENTS
      START       STOP          START        STOP
          0        399              0         399
END OF COPY
END OF JOB
Return value = 0

Copying minidisk 01B4 to 11B4 ...
...
```

The guts of the Redpaper (cont'd)

- Cloning read/write Linux from S10RWMNT to "gold disks"
  - Create a user ID LINUX001

```
USER LINUX001 PASSWD 256M 1G G
 INCLUDE RORDFLT
 OPTION APPLMON
 MDISK 01B0 3390 0001 0030 MM9136 MR PASSWD PASSWD PASSWD
 MDISK 01B1 3390 0031 0400 MM9136 MR PASSWD PASSWD PASSWD
 MDISK 01B4 3390 0431 0550 MM9136 MR PASSWD PASSWD PASSWD
 MDISK 01B5 3390 0981 0100 MM9136 MR PASSWD PASSWD PASSWD
 MDISK 01B6 3390 1081 0400 MM9136 MR PASSWD PASSWD PASSWD
 MDISK 01B7 3390 1481 1750 MM9136 MR PASSWD PASSWD PASSWD
 MDISK 01B8 3390 3231 0108 MM9136 MR PASSWD PASSWD PASSWD
```

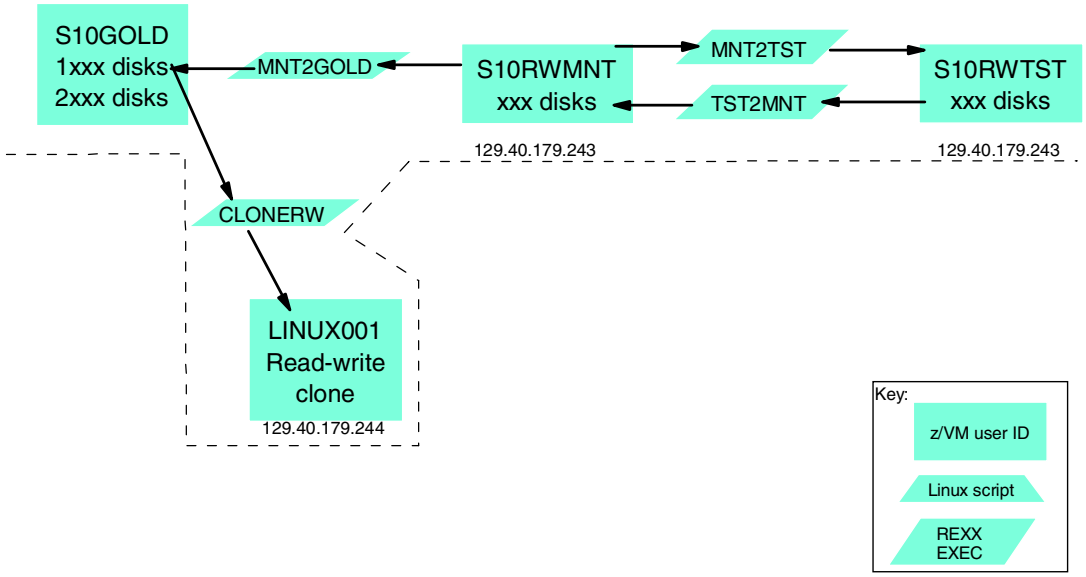
  - Copy a parameter file

```
===> copy s10rwmnt parm-s10 d linux001 = =
===> x linux001 parm-s10 d
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=129.40.179.244 Hostname=ntc244.pbm.ihost.com
...
```

  - Clone to LINUX001

```
==> clonerw linux001
Do you want to copy R/W system from S10GOLD to linux001 y/n
y
Copying minidisk 11B0 to 01B0 ...
...
```


The guts - System block diagram recap



The guts of the Redpaper (cont'd)

- Building a read-only root system
  - Define a user ID, S10ROGLD - all disks R/W
  - Create a prototype system via mkror.sh script then test
    - Guts of the script:

```
sourceID="S10RWMNT"
targetID="S10ROGLD"
rorDiffs="/usr/local/sbin/boot.rootfsck.diffs"
fstabFile="/usr/local/sbin/fstab.ror"

# function calls
checkID $sourceID
checkID $targetID
linkSourceDisks
linkTargetDisks
copySystem
enableSourceDisks
enableTargetDisks
mountSourceRoot
mountTargetDisks
modifySystem
cleanUp
echo "Exiting with rc 0!"
exit 0
```


The guts of the Redpaper (backup)

- Building a read-only root system - function modifySystem()

```
TGT="/mnt/target"
echo "Backing up and modifying /etc/init.d/boot script ..."
cp $TGT/etc/init.d/boot $TGT/etc/init.d/boot.orig
if [ "$?" != 0 ]; then exit 47; fi
cat $TGT/etc/init.d/boot.orig | \
  sed -e 's:bootrc=/etc/init.d/boot.d:bootrc=/sbin/etc/init.d/boot.d:g' > \
  $TGT/etc/init.d/boot
if [ "$?" != 0 ]; then exit 48; fi
echo "Backing up and patching /etc/init.d/boot.rootfsck script ..."
cp $TGT/etc/init.d/boot.rootfsck $TGT/etc/init.d/boot.rootfsck.orig
if [ "$?" != 0 ]; then exit 49; fi
patch $TGT/etc/init.d/boot.rootfsck < $rorDiffs
if [ "$?" != 0 ]; then exit 50; fi
echo "Backing up and copying modified /etc/fstab file ..."
cp $TGT/etc/fstab $TGT/etc/fstab.orig
if [ "$?" != 0 ]; then exit 51; fi
cp $fstabFile $TGT/etc/fstab
if [ "$?" != 0 ]; then exit 52; fi
echo "Backing up and modifying /etc/zipl.conf file ..."
cp $TGT/etc/zipl.conf $TGT/etc/zipl.conf.orig
if [ "$?" != 0 ]; then exit 53; fi
cat $TGT/etc/zipl.conf.orig | \
  sed -e 's/1b0-1bf/1b0(ro),1b1(ro),1b2-1b7,1b8(ro),1b9-1bf/g' > \
  $TGT/etc/zipl.conf
cp $fstabFile $TGT/etc/fstab
if [ "$?" != 0 ]; then exit 54; fi
```

The guts of the Redpaper (cont'd)

- Building a read-only root system
  - Create user ID for R/O clone

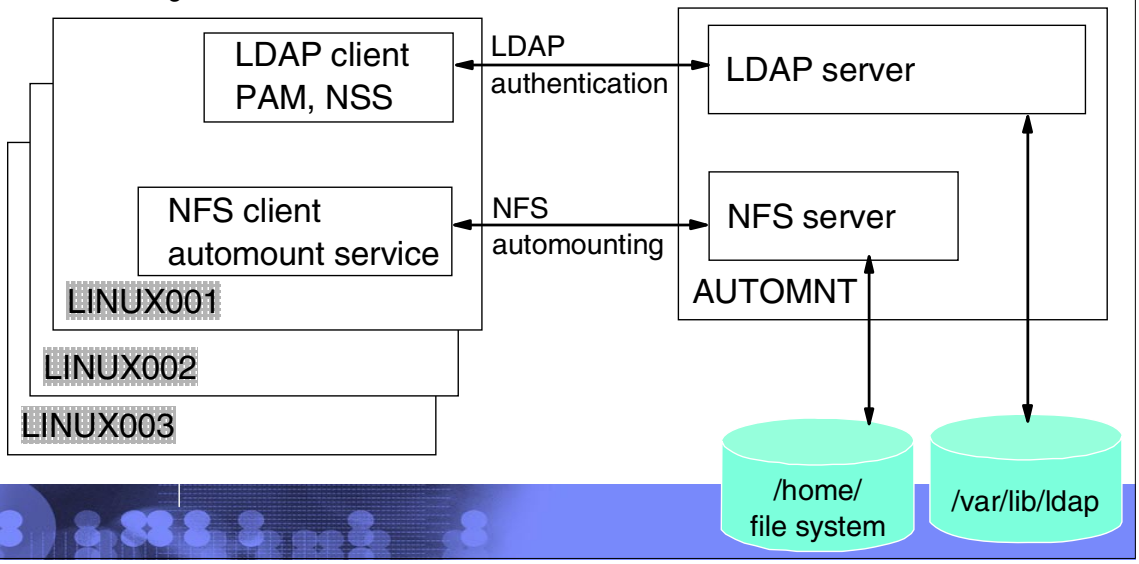
```
USER LINUX002 PASSWD 256M 1G G
 INCLUDE RORDFLT
 OPTION APPLMON
 LINK S10GOLD 11B0 01B0 RR
 LINK S10GOLD 11B1 01B1 RR
 MDISK 01B4 3390 0001 0550 MM9135 MR PASSWD PASSWD PASSWD
 MDISK 01B5 3390 0551 0100 MM9135 MR PASSWD PASSWD PASSWD
 MDISK 01B6 3390 0651 0400 MM9135 MR PASSWD PASSWD PASSWD
 LINK S10GOLD 11B7 01B7 RR
 LINK S10GOLD 11B8 01B8 RR
```

  - Clone a read-only Linux

```
==> clonero linux002
Do you want to copy disks from S10ROGLD to S10GOLD? y/n
y
Copying minidisk 11B0 to 01B0 ...
...
```


Other items in the Redpaper

- Other items
  - Logical Volumes
  - Maintenance of Linux systems
    - Updating the gold disks
    - Updating the cloned servers
  - Travelling /home



Agenda

- ~~Introductions~~
- ~~Overview~~
- ~~Thinking about "systems"~~
- ~~Block diagrams~~
- ~~The "guts" of the paper~~
- Live demo!
- Resources
- Questions

Live Demo

Resources

- Redpaper: "*Sharing and Maintaining Linux under z/VM*"
  - http://www.redbooks.ibm.com/redpieces/abstracts/redp4322.html
- Book: "*z/VM and Linux on System z: The Virtualization Cookbook for {SLES 10|RHEL 5}*"
  - http://linuxvm.org/present/
- *The* Linux for zSeries and S/390 portal
  - http://linuxvm.org/
- The linux-390 list server
  - http://www2.marist.edu/htbin/wlvindex?linux-390
- Linux for zSeries and S/390 developerWorks®
  - http://awlinux1.alphaworks.ibm.com/developerworks/linux390/index.shtml
- SUSE LINUX Enterprise Server 10 SP1 evaluation
  - http://www.novell.com/products/linuxenterpriseserver/eval.html
- z/VM publications
  - http://www.vm.ibm.com/pubs/
- z/VM performance tips
  - http://www.vm.ibm.com/perf/tips/

Questions - ???