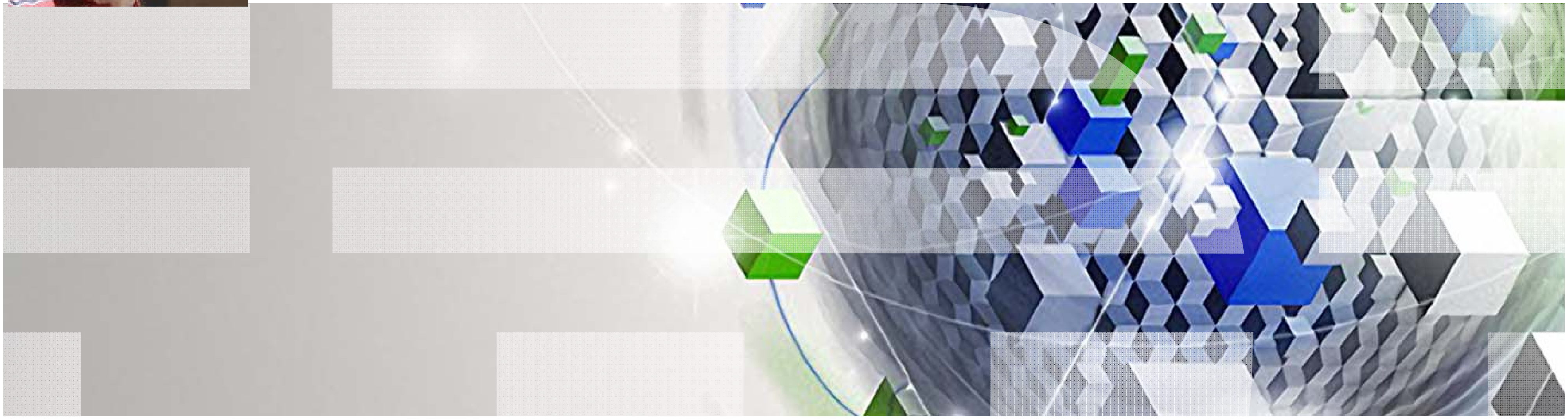
NIM Upgrade Strategies and Advanced Topics

sjknuds@us.ibm.com Steve Knudson

June 2010



Power your planet.

NIM AIX Upgrade Strategies and Advanced Topics

- Migrate 53 to 61 Strategies
- Migrate NIM master first
- Client Migration Methods
  - Migrating from AIX 5.1, 5.2, to 5.3 or 6.1
  - Updates within an AIX Version, TL to TL, or SP to SP
- suma to update NIM lpp_source
- Migrate VIO from 1.5.2.1 to 2.1.0.0
- Various Scripts – “Ideas” for your environment
- Multihomed client – the nimadapters facility

Migrate NIM Master First

- Save NIM database to file in the nimvg (or rootvg if no nimvg)
- mksysb of NIM master rootvg to tape or mkcd to DVD
- alt_disk_copy of master rootvg to unused disks, creating altinst_rootvg
- Make note of physloc(s) for rootvg and altinst_rootvg disks
- Shutdown NIM master, boot with AIX 6.1 media
- Migrate either rootvg or altinst_rootvg disks to AIX 6.1
- All existing NIM resources, machines, networks etc
  still served by 6.1 master, from nimvg filesystems
- Create /export/61 filesystem on nimvg
- smitty bffcreate AIX 6.1 media into /export/61/lppsource_6100-02-02-0849
- smitty nim, point at lpp_source, define as NIM resource
- smitty nim, create spot_6100-02-02-0849 from lpp_source
- smitty nimadm a test client up to AIX 6.1, on unused client disk(s)

Client Migration Methods

- Migrate from older AIX to 5.2, 5.3 or 6.1

•NIM Alternate Disk Migration (nimadm) on a running client (creates altinst_rootvg on unused client disks)
•NIM Alternate Disk Install of new “golden” image (not a migration, a new load on altinst_rootvg)
•mksysb migration – install downlevel mksysb, then migrate it

- Or Update within an AIX Version

•nim_alt_clone with update_all (from TL to TL, or SP to SP)
•multibos – standby copy of AIX on same rootvg disks (updates optional, TL to TL, SP to SP)

NIM Alternate Disk Migration (nimadm 1 of 3)

- At nim master, smitty nimadm
- At nim client, running, existing rootvg clones itself onto unused disk(s) as altinst_rootvg
- Running client then nfs exports altinst_rootvg filesystems for nim master to mount, and migrate to new Version of AIX
- Bootlist on running client optionally set to boot from migrated altinst_rootvg disks
- Boot from upgraded disk at your convenience, verify behavior, if satisfactory stay there. If not, set bootlist back to old_rootvg disks.

NIM Alternate Disk Migration (nimadm 2 of 3)

```
smitty nimadm
                      Perform NIM Alternate Disk Migration

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                   [Entry Fields]
* Target NIM Client                                  [nimclient]            +
* NIM LPP_SOURCE resource                            [lppsource_53ML2]      +
* NIM SPOT resource                                  [spot_53ML2]           +
* Target Disk(s) to install                          [unused hdisks in client]
  DISK CACHE volume group name                       []

  NIM IMAGE_DATA resource                            []                     +
  NIM BOSINST_DATA resource                          []                     +
  NIM EXCLUDE_FILES resource                         []                     +
  NIM INSTALLP_BUNDLE resource                       []                     +
  NIM PRE-MIGRATION SCRIPT resource                  []                     +
  NIM POST-MIGRATION SCRIPT resource                 []                     +

  Phase to execute                                   [all]                  +
  NFS mounting options                               []
  Set Client bootlist to alternate disk?              yes                   +
  Reboot NIM Client when complete?                    no                    +
  Verbose output?                                     no                    +
  Debug output?                                       no                    +

  ACCEPT new license agreements?                      yes                   +
```

NIM Alternate Disk Migration (nimadm 3 of 3)

- nimadm can be finicky...
  - -You must install bos.alt_disk_install.rte into the spot (smitty nim_res_op, cust operation on the spot, install from the lppsource)
  - –An AIX 5.3 NIM master can nimadm a 433 client to 5.1 or 5.2 or 5.3, if you install bos.alt_disk_install.rte.5.3 into the 5.1 and 5.2 spots. But do not copy it into the 5.1 and 5.2 lppsource.
  - –bos.alt_disk_install.rte - same level installed on master as level installed in the spot
- But the error messages are generally coherent

NIM Alternate Disk Install (1 of 1)

- When the running rootvg image is not migrated forward but a new "golden" (e.g., 5.3 or 6.1) image is desired

- Build new golden image on test system

- Pull golden mksysb image from test system to NIM master

- Use NIM to drive golden mksysb image onto unused disks on NIM client. Those disks become altinst_rootvg, while running AIX instance remains on rootvg.

- At maintenance window, boot from altinst_rootvg and verify behavior. If satisfactory, remain on new image that is now labeled rootvg, previous image is labeled old_rootvg

mksysb migration (1 of 3)

- Allows “restore” of downlevel, non-supported mksysb onto Power5 or Power6 hardware, followed by immediate migration to AIX 53TL6.
- Avoids requirement to upgrade existing system to newer Version of AIX before taking mksysb.
- Network boot target LPAR / System with 53TL6 NIM spot
- Restore downlevel mksysb (as far back as 433)
- Immediate Migration of restored mksysb to 53TL6
- See “NIM From A to Z in AIX 5L” pp 205-216, SG24-7296 http://www.redbooks.ibm.com/abstracts/sg247296.html?Open
- Also documented in AIX 5.3 Installation Guide

mksysb migration (2 of 3)

- No BOS menu support of mksysb migration – no prompt required
- bosinst_data file requires the following fields

```
INSTALL_METHOD = migrate
EXISTING_SYSTEM_OVERWRITE = yes
PROMPT = no
RECOVER_DEVICES = no
MKSYSB_MIGRATION_DEVICE = network

target_disk_data:
 PVID =
 PHYSICAL_LOCATION = U9111.520.10C1C1C-V9-C5-T1-L810000000000
 CONNECTION =
 LOCATION =
 SIZE_MB =
 HDISKNAME =
```

New and important field

Physloc is the most definite identifier for target disk. You may have to network boot the target to read these from the menus, then put them into bosinst_data file on master, and boot the target again

mksysb migration (3 of 3)

- You cannot nim –o bosinst (smitty nim_bosinst) a mksysb install to a target with mksysb at 5.2 and spot at 5.3. This will error.

- On the master, pre-allocate the downlevel mksysb to the target client
  nim –o allocate –a mksysb=image_52ML9 <nimclient>

- Then on the master…
  smitty nim_bosinst / rte install / spot_53TL6 / lppsource_53TL6/ bosinst_data / etc

- And then netboot the client. 5.2 mksysb image restores, and is immediately migrated to 5.3.

nim_alt_clone with update_all (1of 2)

- Update within same Version/Release (e.g. 53TL4SP2 to 53TL5)
  - From one TL to next TL
  - From one SP to next SP on same TL

- Build lpp_source at desired level, with simages=yes attribute

- Update NIM master from this lpp_source, reboot master

- If client is mirrored, should unmirror prior to nim_alt_clone:
  - unmirrorvg rootvg hdisk1

    message about quorum being on for rootvg at next reboot, quorum off currently, safe to ignore at this point
  - chpv –c hdisk1
  - reducevg rootvg hdisk1
  - bosboot –a
  - bootlist –m normal hdisk0 (the disk remaining in rootvg)

nim_alt_clone with update_all (2 of 2)

- At the master, smitty nim_alt_clone
- Specify target client (F4 to select)
- Specify target disks (you may select on the running client, say, hdisk1)
- Specify “FIXES to Install” as “update_all”
- Specify “LPP_SOURCE” (F4 to select lpp_source from above)
- Accept Licenses
- Clone and update takes place onto altinst_rootvg, while running on rootvg
- At maintenance window, boot from altinst_rootvg, verify behavior
- If acceptable, stay on new rootvg, if not, boot back to old_rootvg
- If desired mirror new rootvg
  - *might* alt_disk_install –X (clear old_rootvg off hdisk0)
  - extendvg rootvg hdisk0
  - mirrorvg rootvg (msg quorum off on next boot – important!)
  - bosboot –a
  - bootlist –m normal hdisk1 hdisk0
  - shutdown –Fr (to have quorum off on next varyonvg rootvg)

Multibos Support (1 of 6)

- SCSI disks continue to get larger.

- Must I have another unused 70GB disk to alt_disk_clone? No longer

- AIX 53ML2 introduced multibos command. Can create two bootable AIX rootvgs on a single hdisk, or mirrored hdisks

- Defaults to copy / /usr /var /opt /home filesystems. Others can be added to this list, or left to be shared between the two OS images.

- Many flags on the multibos command. See the man page http://publib.boulder.ibm.com/infocenter/pseries/topic/com.ibm.aix.doc/cmds/aixcmds3/multibos.htm

- Some important examples:

  - Before you start, you might extendlv hd5 from one logical partition to two contiguous partitions
    lslv –m hd5
    extendlv hd5 1
    bosboot –a –d hdiskn

  - Create standby copy of OS – multibos –Xs
  - Add updates to standby copy from /images – multibos –Xac –l /images

Multibos Support (2 of 6)

- Log file defaults to /etc/multibos/logs/op.alog – sample output here:

```
===============================================================================
DATE: 2005.11.08.10:54:56 ID: [3 5 00CC1C1C4C00] COMMAND: (multibos -Xsp)
===============================================================================
Gathering system information ...

+-----------------------------------------------------------------------------+
Preview
+-----------------------------------------------------------------------------+
Verifying operation parameters ...
Processing preview information ...

ACTIVE LV:          hd4
STANDBY LV:         bos_hd4
TYPE:               jfs2
ACTIVE FS:          /
STANDBY FS:         /bos_inst
ACTION:             Setup
STATE:              mounted
```

Multibos Support (3 of 6)

- More log output…

```
+-----------------------------------------------------------------------------+
Logical Volumes
+-----------------------------------------------------------------------------+
Creating standby BOS logical volume bos_hd5
Creating standby BOS logical volume bos_hd4
Creating standby BOS logical volume bos_hd2
Creating standby BOS logical volume bos_hd9var
Creating standby BOS logical volume bos_hd1
Creating standby BOS logical volume bos_hd10opt

+-----------------------------------------------------------------------------+
File Systems
+-----------------------------------------------------------------------------+
Creating all standby BOS file systems ...
```

Multibos Support (4 of 6)

- Check the bootlist after multibos command:

```
# bootlist -m normal –o
hdisk0 blv=bos_hd5
hdisk0 blv=hd5
```

- Change bootlist if desired:

```
# bootlist -m normal hdisk0 blv=hd5 hdisk0 blv=bos_hd5
```

- Check bootlist again:

```
# bootlist -m normal –o
hdisk0 blv=hd5
hdisk0 blv=bos_hd5
```

Multibos Support (5 of 6)

➢At the end of the multibos command output,

```
Setting bootlist to logical volume bos_hd5 on hdisk0.
 ATTENTION: firmware recovery string for standby BLV (bos_hd5):
 boot /vdevice/v-scsi@30000005/disk@8100000000000000:4
 ATTENTION: firmware recovery string for active BLV (hd5):
 boot /vdevice/v-scsi@30000005/disk@8100000000000000:2

 Log file is /etc/multibos/logs/op.alog
 Return Status = SUCCESS
```

Note these values and corresponding boot lvs

- ➢ Do both of these boot devices both show up in SMS? Of course!

Multibos Support (6 of 6)

SMS Selecting boot / install device, scsi disk

```
  PowerPC Firmware
Version SF235_185
SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.
-----------------------------------------------------------------------
  --
Select Device
Device  Current  Device
Number  Position  Name
1.        2      SCSI 13421 MB Harddisk, part=2 (AIX 5.3.0)
                 ( loc=U9111.520.10C1C1C-V2-C5-T1-W810000000
2.        1      SCSI 13421 MB Harddisk, part=4 (AIX 5.3.0)
                 ( loc=U9111.520.10C1C1C-V2-C5-T1-W810000000
```

part=2, part=4 from multibos output

Same LUN, W81000000, one hdisk

- ➢ Two copies on one disk doesn’t protect against a spindle failure. But does protect against operator errors (e.g., rm –r /usr)

Let’s NOT try this sample syntax

SUMA Examples

- Update an lppsource

```
suma -x -a RqType=SP –a RqName=5300-07-06-0844 -a Action=Preview \
-a FilterDir=/export/53/lppsource_53TL7 -a DLTarget=/export/53/lppsource_53ML7
suma -x -a RqType=SP –a RqName=5300-08-04-0844 -a Action=Download \
-a FilterDir=/export/53/lppsource_53ML7 -a DLTarget=/export/53/lppsource_53ML7
```

Note that pointing at the lppsource directory puts the updates down in <lppsource>/installp/ppc
Same code, may be packaged in different SP for different TL

- Update a running client

```
  lslpp -Lc >./list
  suma -x -a RqType=Security -a Action=Preview -a FilterSysFile=./list -a DLTarget=/usr/sys/inst.images
  suma -x -a RqType=Critical -a Action=Preview -a FilterSysFile=./list -a DLTarget=/usr/sys/inst.images
  suma -x -a RqType=APAR -a RqName=IY12345 -a Action=Preview -a FilterSysFile=./list \
    -a DLTarget=/usr/sys/inst.images
suma -x -a RqType=TL -a RqName=5300-08 -a Action=Preview -a FilterSysFile=./list \
    -a DLTarget=/usr/sys/inst.images
  suma -x -a RqType=Fileset -a RqName=bos.rte.lvm -a Action=Preview -a FilterSysFile=./list  \
    -a DLTarget=/usr/sys/inst.images
```

change Action from Preview to Download to actually download the fixes

- I occasionally use suma to update SP or TL of an older lpp_source
- Better to build new lpp_source from new media – will not miss “new” base install filesets

SUMA Examples

- If your network requires use of proxy to reach public Internet, you may add proxy config:

  Bypass SUMA's proxy handling by clearing the HTTPS_PROXY value

  ```
  suma -ca HTTPS_PROXY=
  ```

  Set environment variables instead as follows:

  ```
  export HTTPS_PROXY="<proxy>"
  ```

  and, if appropriate:

  ```
  export HTTPS_PROXY_USERNAME="<username>“
  export HTTPS_PROXY_PASSWORD="<password>“
  ```

- Or add one of these right on the suma command line:

  ```
  -a HTTP_PROXY=http://proxy.your.domain.com:8080
  -a HTTP_PROXY=http://username:password@proxy.your.domain.com:8080
  ```

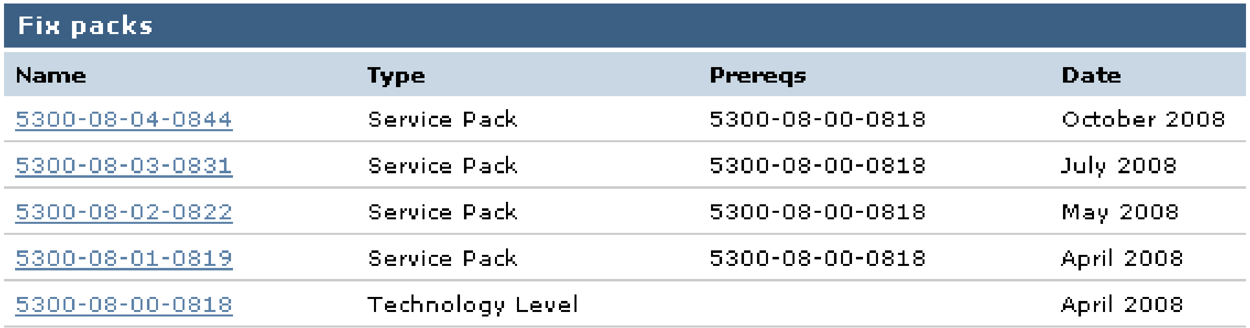
- Or, each of these might be required on the command line

  ```
  -a FIXSERVER_PROTOCOL=https
  -a DOWNLOAD_PROTOCOL=https
  -a HTTPS_PROXY=https://username:password@proxy.your.domain.com:443
  ```

- Hint: Support Line has worked a number of these

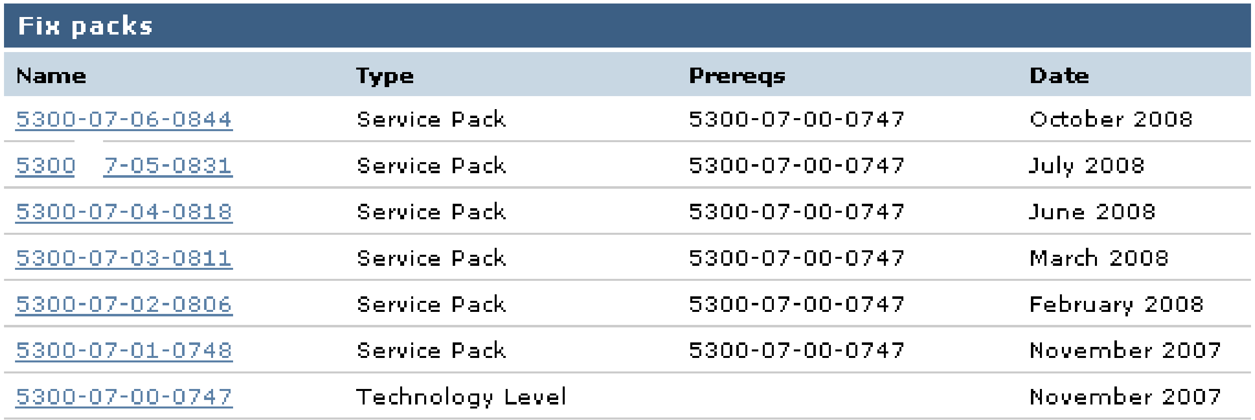
SUMA Examples

- ➢Can you select equivalent code at different Technology Levels? Yes
- ➢http://www-933.ibm.com/eserver/support/fixes/fixcentral/pfixpacks/53



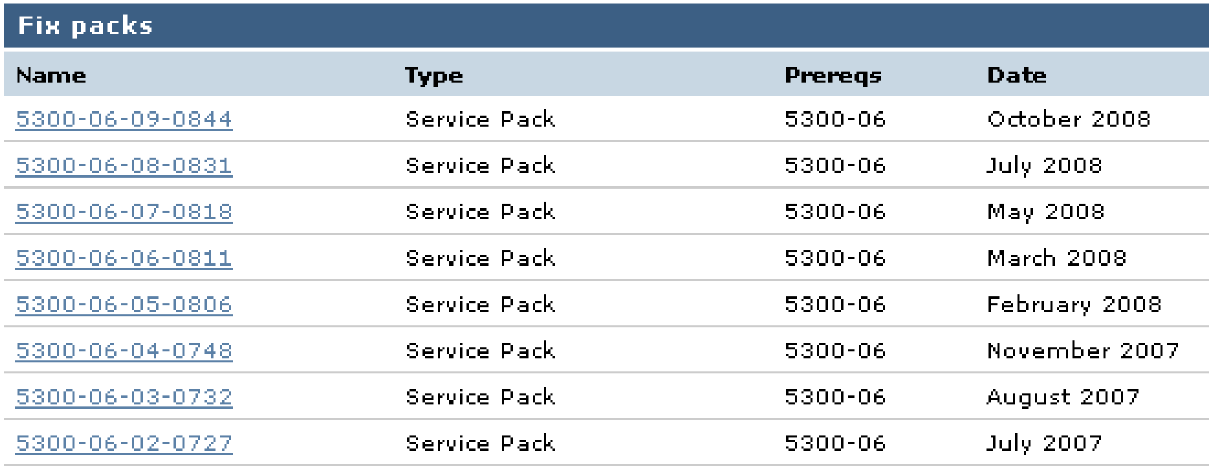
Fix packs

Name	Type	Prereqs	Date
5300-08-04-0844	Service Pack	5300-08-00-0818	October 2008
5300-08-03-0831	Service Pack	5300-08-00-0818	July 2008
5300-08-02-0822	Service Pack	5300-08-00-0818	May 2008
5300-08-01-0819	Service Pack	5300-08-00-0818	April 2008
5300-08-00-0818	Technology Level		April 2008



Fix packs

Name	Type	Prereqs	Date
5300-07-06-0844	Service Pack	5300-07-00-0747	October 2008
5300 7-05-0831	Service Pack	5300-07-00-0747	July 2008
5300-07-04-0818	Service Pack	5300-07-00-0747	June 2008
5300-07-03-0811	Service Pack	5300-07-00-0747	March 2008
5300-07-02-0806	Service Pack	5300-07-00-0747	February 2008
5300-07-01-0748	Service Pack	5300-07-00-0747	November 2007
5300-07-00-0747	Technology Level		November 2007



Fix packs

Name	Type	Prereqs	Date
5300-06-09-0844	Service Pack	5300-06	October 2008
5300-06-08-0831	Service Pack	5300-06	July 2008
5300-06-07-0818	Service Pack	5300-06	May 2008
5300-06-06-0811	Service Pack	5300-06	March 2008
5300-06-05-0806	Service Pack	5300-06	February 2008
5300-06-04-0748	Service Pack	5300-06	November 2007
5300-06-03-0732	Service Pack	5300-06	August 2007
5300-06-02-0727	Service Pack	5300-06	July 2007

Migrate VIO from 1.5.2.1 to 2.1.0.0

Boot Migration Media

- Preserves SEA config, and vhost virtual disk mapping
- Watch for some fileset microsurgery
  bos.INed 6.1.2.0
  bos.INed 6.1.0.0
  bos.mls.rte 6.1.2.0
  devices.pciex.b3154a63.rte 6.1.0.0 PCI-E DDR Infiniband driver
  devices.vdevice.IBM.vfc-client.rte 6.1.0.0 Virtual Fibre Channel Client Support
- Doubt anyone will miss INed
- Found Infiniband driver at 5.3.8.1 on migrated VIO, but at 6.1.2.0 plus 6.1.2.2 on scratch 2.1 VIO install
- Found VFC at 6.1.2.0 plus 6.1.2.2 on a new scratch VIO 2.1 install, but not on a migrated VIO
- VIO 2.1 migration DVD iso image –
  http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/home.html

Migrate VIO from 1.5.2.1 to 2.1.0.0

Phase 2 – Really do need a NIM solution for this migration – can fix missing filesets in the process

- copied <cdrom>/installp/ppc to NIM master, defined it as lpp_vio21_mig lpp_source

- copied into this lpp_source, from 6100-02-02-0849 lppsource, the following:
  devices.chrp.IBM.HPS.1.4.0.0.I (instfix –icqk 6100-02_AIX_ML | grep ":-:" or, oslevel -rl)
  devices.common.IBM.sni.1.4.0.0.I
  devices.msg.en_US.chrp.IBM.HPS.1.4.0.0.I
  devices.msg.en_US.common.IBM.sni.1.4.0.0.I
  devices.pciex.8680c71014108003.6.1.2.0.I
  devices.pciex.b3154a63.6.1.2.1.I

- did initial scratch install of VIO server with VIO 1521 mksysb and spot

- copied VIO migration <cdrom>/bosinst.data file to NIM master, modified it for migration, defined it as NIM resource

- enabled RTE install of VIO server with lpp_vio21_mig lpp_source, vio21 spot extracted from vio21 mksysb, and migration bosinst_data resource

- Migrated VIO 2.1 now shows same TL as scratch install VIO 2.1

```
# instfix -i | grep ML
    All filesets for 6100-00_AIX_ML were found.
    All filesets for 6100-01_AIX_ML were found.
    All filesets for 6100-02_AIX_ML were found.
```

Various scripts - turn off TCP/IP Services

```
# turn off certain TCP/IP services on client machine (can do this in an fb_script)
     cp /etc/inetd.conf /etc/inetd.conf.original
     cat /etc/inetd.conf.original | sed s/^ftp/#ftp/ | sed s/^telnet/#telnet/ \
                              | sed s/^shell/#shell/ | sed s/^kshell/#kshell/ \
                              | sed s/^login/#login/ | sed s/^klogin/#klogin/ \
                              | sed s/^exec/#exec/   | sed s/^bootps/#bootps/ \
                              | sed s/^ntalk/#ntalk/ >/etc/inetd.conf

     cp /etc/rc.tcpip /etc/rc.tcpip.original
     cat /etc/rc.tcpip.original | \
     sed s/^qpi=30m/#qpi=30m/ | \
     sed s/"^start \/usr\/lib\/sendmail"/"#start \/usr\/lib\/sendmail"/ | \
     sed s/"^start \/usr\/sbin\/snmpd"/"#start \/usr\/sbin\/snmpd"/ | \
     sed s/"^start \/usr\/sbin\/portmap"/"#start \/usr\/sbin\/portmap"/ | \
     sed s/"^start \/usr\/sbin\/dpid2"/"#start \/usr\/sbin\/dpid2"/ >/etc/rc.tcpip

     cp /etc/inittab /etc/inittab.original
     cat /etc/inittab.original | sed s/^rcnfs/:rcnfs/ | \
                                    sed s/^qdaemon/:qdaemon/ | \
                                    sed s/^writesrv/:writesrv/  > /etc/inittab
```

AIX 6.1
SECURE_BY_DEFAULT
option in bosinst.data file
should now handle

Various scripts - get_mksysb

```
#!/bin/ksh
# Copyright IBM Corporation 2002, 2008 All Rights Reserved.
#
usage() {
  echo ""
  echo "get_mksysb [-r] [-n] [-m 'machines to get mksysb from']"
  echo "   "
  echo "    -r remove oldest mksysb for client machine"
  echo "    -n no_make - do not make new mksysb after removing oldest"
  echo "    -m machine_list"
  echo ""
  echo "get mksysb from each client machine specified with -m."
  echo ""
  echo "-r flag says remove oldest existing mksysb for the machines"
  echo "   being backed up.  You must specify machine list with -m flag"
  echo "   when using -r flag."
  echo ""
  echo "-n no_make flag with -r to remove a generation of mksysb, without"
  echo "   creating a new one.  Again, -m flag is required with -r flag."
     echo ""
  echo "script assumes client mksysbs should be placed in a directory"
  echo "/export/mksysb/client.  This will allow mksysb for one client to"
  echo "be made while other clients are enabled for install, without an“
  echo "nfs export conflict"
  exit 1
  }
```

Various scripts - get_mksysb

```
remove_old=

machine_list=

no_make=

while getopts rnh?m: option
   do
   case $option in
   r) remove_old=1;;
   m) machine_list="$OPTARG";;
   n) no_make=1;;
   h) usage;;
   ?) usage;;
   esac
   done

# 3-10-2008, do not run without user specified machine_list
if [ -z "$machine_list" ]; then
 usage
fi

# do not remove mksysb images without a machine_list
if [ -z "$machine_list" ] && [ ! -z "$remove_old" ]; then
   usage
fi

# do not run without some command line arguments
if [ $# -eq 0 ]; then
   usage
fi
```

Various scripts - get_mksysb

```
# if machine_list is null at this point, set it to ALL clients
# As of 3-10-2008, won't do this, because we always expect a user
# specified machine_list
#if [ -z "$machine_list" ]; then
#   machine_list=`lsnim -c machines | grep -v master | awk '{print $1}'`
#fi

# echo machine list is $machine_list
cd /export/mksysb
for m in $machine_list
   do

   if [ ! -z "$remove_old" ]; then
      nim_mksysb_res=$(ls -lt $m  2>>/dev/null | grep $m | tail -1 | awk '{print $9}')
      if [ ! -z "$nim_mksysb_res" ]; then
         echo removing old file and nim resource $nim_mksysb_res
         nim -o remove $nim_mksysb_res
         rm $m/$nim_mksysb_res
      fi
   fi
```

Various scripts - get_mksysb

```
#  if no_make is null, go ahead and make the mksysb
     if [ -z "$no_make" ]; then
        if [[ ! -d "/export/mksysb/$m" ]] then
           mkdir /export/mksysb/$m
        fi
     filename="$m"_`date +%Y%m%d%H%M`
     echo new file / nim resource is $filename
     echo machine to backup is $m
     nim -o define -t mksysb -aserver=master -amk_image=yes \
     -alocation=/export/mksysb/$m/$filename \
     -asource=$m $filename
     fi
  done
```

Various scripts – NIM “script” for rte install

```
#!/bin/ksh

chfs -a size=+1 /
chfs -a size=+1 /home
chfs -a size=+1 /var
chfs -a size=+1 /tmp
chfs -a size=+1 /opt
```

Increase filesystems by one 512 Byte block (essentially, 1physical partition)

```
echo "set -o vi" >>/.profile
echo 'export PS1="`whoami `@`hostname ` \$PWD # "' >>/.profile
echo "export PATH=$PATH:/usr/local/bin" >>/.profile
chmod u+x /.profile
```

Insert content into /.profile

```
if [ `oslevel` == "5.3.0.0" ] ; then
  chdev -l aio0 -a minservers=20 -a maxservers=40 -a autoconfig=available
fi
```

AIO config for AIX 5.3, after install, before boot

```
chuser fsize=-1 root

grep -p -v "^root:" /etc/security/passwd >/etc/security/passwd.noroot
echo "root:" >/etc/security/passwd
echo "\tpassword = IG.NHtgcCnUno" >>/etc/security/passwd
echo "\tlastupdate = `date +%s`" >>/etc/security/passwd
echo "\tflags = " >>/etc/security/passwd
echo "" >>/etc/security/passwd
cat /etc/security/passwd.noroot >>/etc/security/passwd
rm /etc/security/passwd.noroot
```

Set password for root

Various scripts – NIM “script” for rte install

```
chdev -l hdisk0 -a hcheck_interval=300 -P

echo "hosts=local,bind" >>/etc/netsvc.conf

/usr/sbin/mkitab -i rcnfs "sknim:2:once:/usr/local/sknim_bundle >>/smit.log 2>&1 "

mkdir /usr/local

echo '/usr/lpp/bos.sysmgt/nim/methods/c_sm_nim inst_bundle \' >>/usr/local/sknim_bundle
echo '-l lppsource_ssl_ssh \' >>/usr/local/sknim_bundle
echo '-b "ssl_53_bundle" \' >>/usr/local/sknim_bundle
echo '-f Y -f c -f N -f g -f X ' >>/usr/local/sknim_bundle
echo 'wait' >>/usr/local/sknim_bundle
echo "" >>/usr/local/sknim_bundle

echo '/usr/lpp/bos.sysmgt/nim/methods/c_sm_nim inst_bundle \' >>/usr/local/sknim_bundle
echo '-l lppsource_ssl_ssh \' >>/usr/local/sknim_bundle
echo '-b "ssh_53_bundle" \' >>/usr/local/sknim_bundle
echo '-f Y -f c -f N -f g -f X ' >>/usr/local/sknim_bundle
echo 'wait' >>/usr/local/sknim_bundle
echo "" >>/usr/local/sknim_bundle
```

Set MPIO before boot, to check failed or non-active paths every 300 sec

Make inittab entry after nfs startup, for script we create below

Various scripts – NIM “script” for rte install

```
echo 'no -p -o rfc1323=1' >>/usr/local/sknim_bundle
echo 'no -p -o tcp_sendspace=262144' >>/usr/local/sknim_bundle
echo 'no -p -o tcp_recvspace=262144' >>/usr/local/sknim_bundle
```

Set some no options for now and restart

```
echo 'if [ `oslevel` == "6.1.0.0" || `oslevel` == "6.1.2.0" ] ; then' >>/usr/local/sknim_bundle
echo '   ioo -p -o aio_minservers=20' >>/usr/local/sknim_bundle
echo '   ioo -p -o aio_maxservers=40' >>/usr/local/sknim_bundle
echo '   ioo -p -o posix_aio_minservers=20' >>/usr/local/sknim_bundle
echo '   ioo -p -o posix_aio_maxservers=40' >>/usr/local/sknim_bundle
echo 'fi' >>/usr/local/sknim_bundle
```

AIO config for AIX 6.1 has changed

```
echo ‘/usr/sbin/rmtcpip’ >>/usr/local/sknim_bundle
```

The IP that this client installed with may be used on another target machine. Must rmtcpip at the end of processing here. Would need more intelligent selection of interface to unconfigure if multiple interfaces are configured with nimadapters

```
echo 'rmitab sknim' >>/usr/local/sknim_bundle
chmod u+x /usr/local/sknim_bundle
```

Script removes itself from inittab at end of execution

cron script on master keeps addresses enabled

```
 #!/bin/ksh
#
# sknim_bosinst - goes thru the machine list, if Cstate is "ready for nim
#               operation", clear the cpuid from the last install and
#               do the bosinst operation to renable the machine for install
#
#
#for machine in ec01 ec02 ec10 ec11
for machine in sq17 sq18
   do
   if lsnim -l $machine | grep "Cstate        = ready for a NIM operation" >/dev/null ; then
      echo enabling $machine ...
      nim -o change -a cpuid="" $machine
      wait ;
      nim -o bos_inst -a source=mksysb -a spot=spot_vio \
                      -a bosinst_data=bosinst_vio \
                      -a mksysb=mksysb_vio \
                      -a resolv_conf=resolv_51 \
                      -a accept_licenses=yes \
                      -a no_client_boot=yes -a preserve_res=yes \
                      -a installp_flags=cNgXY $machine
  fi
  done
```

These clients are always enabled for a VIO install

cron script on master keeps addresses enabled

These clients are
always enabled for
a AIX 6 install

```
for machine in ec03 ec04
  do
  if lsnim -l $machine | grep "Cstate       = ready for a NIM operation" >/dev/null ; then
    echo enabling $machine ...
    nim -o change -a cpuid="" $machine
    wait ;
    nim -o bos_inst -a source=rte -a spot=spot_aix6 \
                -a lpp_source=lpp_source_aix6 \
                -a bosinst_data=bosinst_53_64bit \
                -a script=order_52_script \
                -a resolv_conf=resolv_51 \
                -a accept_licenses=yes \
                -a no_client_boot=yes -a preserve_res=yes \
                -a installp_flags=cNgXY $machine
  fi
  done
```

Various scripts - reset client machine state

```
#!/bin/ksh
#
# to reset nim state of a nim client, not running
#
# sknim_reset client
#
nim -o reset -aforce=yes $1
nim -Fo deallocate -asubclass=all $1
nim -o change -a cpuid="" $1
```

Multihomed NIM Client

- NIM master and client environment
- NIM adapters resource
- Client secondary adapters stanza file
- nimadapters command
- Install of client
- Check client config after install
- Configure Client nimsh after install
- Set client hostname, nimsh config
- NIM master to client activities
  - Preview update_all
  - Collect mksysb from client
  - HMC DLPAR memory, virtual adapter into client
- EtherChannel by nimadapters

NIM master and client environment

- NIM master fahr.dfw.ibm.com 9.19.51.115 oslevel –s 6100-02-02-0849
- NIM client bmark28.dfw.ibm.com 9.19.51.229 oslevel –s 5300-09-02-0849

- Client needs following interfaces

en0 9.19.51.228 bmark28	HMC and NIM (admin network)
en1 129.1.1.228 bmark28_129	Public facing
en2 172.16.1.228 bmark28_172	TSM
en3 192.168.16.228 bmark28_192	Oracle HSI

- Objectives
  install thru en0 NIM interface
  configure all interfaces on install
  configure client communications services for nimsh
  set client hostname to match public facing interface (bmark28_129)
  preserve “all” nimsh functions from master to client (preview update_all, collect mksysb. 1Q2009, nimadm still requires rsh)

NIM Adapters resource

- NIM “adapter_def” resource is a directory that holds client secondary adapter definitions

- `# mkdir /export/adapters`
- smitty nim / perform nim adminstration tasks / Manage Resources / Define a Resource / type adapter_def

```
                        Define a Resource

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                     [Entry Fields]
* Resource Name                                      [adapters]
* Resource Type                                       adapter_def
* Server of Resource                                 [master]               +
* Location of Resource                               [/export/adapters]      /
  NFS Client Security Method                         []                     +
  NFS Version Access                                 []                     +
  Comments                                           []
```

NIM Adapters resource

```
# lsnim -l adapters
adapters:
   class       = resources
   type        = adapter_def
   Rstate      = ready for use
   prev_state  = unavailable for use
   location    = /export/adapters
   alloc_count = 0
   server      = master
```

Client Secondary Adapters stanza file

Many options – see “man nimadapters”

```
# pg /export/res/bmark28.adapter
default:
        machine_type = secondary
        subnet_mask = 255.255.255.0
        network_type = en
        cable_type = N/A

bmark28:
        netaddr = 129.1.1.228
        interface_name = en1
        secondary_hostname = bmark28_129
        route="0::::129.1.1.1"

bmark28:
        netaddr = 172.16.1.228
        interface_name = en2
        secondary_hostname = bmark28_172

bmark28:
        netaddr = 192.168.16.228
        interface_name = en3
        secondary_hostname = bmark28_192
(EOF):
```

Stanza file is not in the adapter_def directory

Default settings for all adapters following. These attributes can be overridden in adapter stanza

Wanted a default route on this one… didn’t get it – maybe if I take the default route off the NIM network definition on the master

You may specify “location” instead of “interface_name”. See man nimadapters

nimadapters command

Generate secondary adapter definitions from the stanza file

```
# nimadapters –d –f /export/res/bmark28.adapters adapters
```

```
# pg /export/adapters/bmark28.adapters
bmark28:
   hostname=bmark28
   machine_type=secondary
   network_type=en
   hostaddr=9.19.51.228
   secondary_hostname=bmark28_129
   netaddr=129.1.1.228
   subnet_mask=255.255.255.0
   cable_type=N/A
   interface_name=en1
   media_speed=Auto_Negotiation
   route="0::::129.1.1.1"
bmark28:
   hostname=bmark28
   machine_type=secondary
   network_type=en
   hostaddr=9.19.51.228
   secondary_hostname=bmark28_172
   netaddr=172.16.1.228
   subnet_mask=255.255.255.0
   cable_type=N/A
   interface_name=en2
   media_speed=Auto_Negotiation
```

```
bmark28:
   hostname=bmark28
   machine_type=secondary
   network_type=en
   hostaddr=9.19.51.228
   secondary_hostname=bmark28_192
   netaddr=192.168.16.228
   subnet_mask=255.255.255.0
   cable_type=N/A
   interface_name=en3
   media_speed=Auto_Negotiation
```

You might even be able to just edit these in

Install of client

```
# nim –o allocate –a adapter_def=adapters bmark28
# nim -o bos_inst -a source=rte -a spot=spot_53TL9 \
                    -a lpp_source=lppsource_5300-09-02-0849 \
                    -a bosinst_data=bosinst_53_64bit \
                    -a script=order_52_script \
                    -a resolv_conf=resolv_51 \
                    -a accept_licenses=yes \
                    -a no_client_boot=yes -a preserve_res=yes \
                    -a installp_flags=cNgXY bmark28
```

Network boot client, and perform rte install of AIX

Check client config after install

```
# netstat –in
Name  Mtu   Network     Address              Ipkts Ierrs    Opkts Oerrs  Coll
en0   1500  link#2      4e.c4.33.88.21.b    497543     0   930053     0     0
en0   1500  9.19.51     9.19.51.228         497543     0   930053     0     0
en1   1500  link#3      4e.c4.33.88.21.c         0     0       14     0     0
en1   1500  129.1.1     129.1.1.228              0     0       14     0     0
en2   1500  link#4      4e.c4.33.88.21.d         0     0        4     0     0
en2   1500  172.16.1    172.16.1.228             0     0        4     0     0
en3   1500  link#5      4e.c4.33.88.21.e         0     0        4     0     0
en3   1500  192.168.16  192.168.16.228           0     0        4     0     0
lo0   16896 link#1                             294     0      314     0     0
lo0   16896 127         127.0.0.1              294     0      314     0     0
lo0   16896 ::1
```

- Did not get my default route 129.1.1.1, had to add it – smitty route. My HMC and NIM master are on the same 9.19.51.0 network; no route added for these.

```
# tail -5 /etc/hosts
9.19.51.228     bmark28.dfw.ibm.com
9.19.51.115     fahr.dfw.ibm.com
129.1.1.228     bmark28_129
172.16.1.228    bmark28_172
192.168.16.228  bmark28_192
```

Did get the secondary hostnames in /etc/hosts

Set client hostname, nimsh config

# smitty hostname, set to bmark28_129

# smitty nim / Configure Client Communication Services

```
 Configure Client Communication Services

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                  [Entry Fields]
* Communication Protocol used by client             [nimsh]                 +

  NIM Service Handler Options
*   Enable Cryptographic Authentication             [disable]               +
      for client communication?

    Install Secure Socket Layer Software (SSLv3)?   [no]                    +
      Absolute path location for RPM package        [/dev/cd0]               /
                  -OR-
      lpp_source which contains RPM package         []                      +

    Alternate Port Range for Secondary Connections
       (reserved values will be used if left blank)
```

Still battling for the right combo of openssl, openssh. Was able to use nimsh without ssl encryption

NIM master to client activities

- Successfully previewed an update_all from master to client bmark28, while it was wearing hostname bmark28_129

- Successfully collected mksysb image from bmark28, using get_mksysb script

- Successful DLPAR at HMC to add memory to client LPAR bmark28, also added virtual adapter

EtherChannel by nimadapters

- DLPAR the two virtual Ethernet adapters into the client LPAR ec04. I knew by slot numbers they would configure as ent1 and ent2.

- in client LPAR ec04,

```
cfgmgr
lsdev -Cc adapter -F "name physloc" | grep ent[0-9]    (shows new adapters ent1 and ent2 in the expected slots)
```

- on the nim master, added this to my nimadapters stanza (/export/res/client.adapters) file

```
 ec04:
        machine_type = etherchannel
        network_type = en
        interface_name = en3
        netaddr = 192.168.18.4
        subnet_mask = 255.255.255.0
        adapter_attributes = "adapter_names=ent1 backup_adapter=ent2 mode=standard"
```

- I knew that ent1 and ent2 would make an EtherChannel device ent3, and I specified the IP address for en3 interface

EtherChannel by nimadapters

- on the nim master

```
nimadapters -d -f /export/res/client.adapters
```

This command generated the file ec04.adapter in my adapters resource

```
ec04:
    hostname=ec04
    machine_type=etherchannel
    network_type=en
    hostaddr=9.19.51.82
    netaddr=192.168.18.4
    subnet_mask=255.255.255.0
    cable_type=N/A
    interface_name=en3
    media_speed=Auto_Negotiation
    adapter_attributes="adapter_names=ent1 backup_adapter=ent2 mode=standard"
```

- at the master

```
nim -o cust -a adapter_def=adapters ec04
```

This command configured the EtherChannel ent3 in the client LPAR ec04, and put the IP address on it. I expect it to work in equivalent manner during a nim install.

EtherChannel by nimadapters

Another example client adapters stanza, using physical adapters

```
sq14_192:
        machine_type = etherchannel
        network_type = en
        cable_type = tp
        interface_name = en3
        netaddr = 192.168.1.14
        subnet_mask = 255.255.255.0
        multiple_physloc = U7311.D20.1059EAB-P1-C06-T1,U7311.D20.1059EAB-P1-C06-T2
        media_speed = 100_Full_Duplex
        adapter_attributes = "mode=standard hash_mode=src_dst_port
```

# nimadapters –p –f /export/res/client.adapters (to preview)

# nimadapters –d –f /export/res/client.adapters adapters (put the definition file in my “adapters” resource)

References

- NIM From A to Z in AIX 5L
  http://www.redbooks.ibm.com/abstracts/sg247296.html?Open

- AIX 5L Version 5.3 Installation and Migration
  http://publib.boulder.ibm.com/infocenter/pseries/v5r3/topic/com.ibm.aix.install/doc/insgdrf/insgdrf.pdf

- Installing AIX in a Partitioned Environment
  http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.install/doc/insgdrf/lpar_installing_aix.htm

- Optimizing AIX 6.1 performance tuning
  http://www.ibm.com/developerworks/aix/library/au-aix6tuning/?S_TACT=105AGY06&