PATCHING MIRRORED SYSTEMS WITH THE SOLARIS™ LIVE UPGRADE SOFTWARE

Jeff Smith, Sun Microsystems

Sun BluePrints™ On-Line — June 2007

Table of Contents

Patching Mirrored Systems with the Solaris™ Live Upgrade Software

Introduction

Applying operating system patches is often time consuming and disruptive. In fact, most patching techniques require systems and disks to be taken offline for the patching process. The Solaris™ Live Upgrade software enables administrators to stage patches without impacting running systems, and minimize the actual downtime associated with patching to the time needed to reboot the system. This Sun BluePrints™ article discusses how the Solaris Live Upgrade software can be used to patch mirrored systems and rollback to the pre-patched environment should the need arise. While this article focuses on Sun systems incorporating SPARC® processors that run the Solaris 8, 9, and 10 Operating System (OS), the procedures can be used on x86 and x64 systems as well. However, device naming conventions may be different on these systems. It is important to note that the procedures outlined in this document assume knowledge of the Solaris Volume Manager software and experience with patching the Solaris Operating System on Sun servers.

Solaris Live Upgrade

Solaris Live Upgrade software enables the operating system to continue to run while upgrades, patch installations, or routine maintenance operations are performed. System administrators can patch a system image rapidly without impacting the boot environment needed by the Solaris OS to run. Administrators simply create a copy of the active boot environment, make changes to the copy, and reboot to the updated boot envrenment when appropriate. In the event of a problem, administrators can revert to a previous environment with a simple reboot. The result — a simplified way to update systems that minimizes the downtime and risk often associated with patching efforts.

The use of Solaris Live Upgrade software provides a host of benefits, including:
- Reduces patching downtime to the time it takes to reboot the system
- Requires only a reboot to roll back to the pre-patched state
- Eliminates the need to split mirrors in order to patch the system
- Enables patching to be accomplished during normal business hours
- Provides a simple disaster recovery process if the root volume is corrupted
- Eases the configuration and administration process

Patching Considerations

Several factors should be considered prior to patching a system, including how the patching effort can affect system availability and the risks associated with errant processes.

System Availability

The procedures outlined in sections that follow are designed to maximize availability. After creating a copy of the boot environment, including the /, /var, and /opt directories, the copy is patched rather than the running environment. Patching typically involves applying the Enterprise Installation Services (EIS) patches or Recommended Patch Cluster to a system, and often requires the system run in single user mode for the duration of the patch process. Using the Solaris Live Upgrade software, patches are applied to a copy of the boot environment as a background task on the production server. The only downtime experienced is the time needed to reboot the system when it is time to switch over to the patched boot environment.

Risk Management

Use of the Solaris Live Upgrade software results in two boot environments — the original and patched systems. Once patching is complete, administrators typically reboot the patched environment and verify applications run properly. If problems arise, the fastest recovery path is to reboot the system using the original boot environment. The ability to switch between the changed and unchanged environments offers a useful way to manage the risks associated with making alterations to systems.

Caveats

There are two primary issues to consider when using the Solaris Live Upgrade software for patching.

- Patches must not make changes to the current boot environment. While Sun patches adhere to these rules, third-party patches may behave differently.

- This document specifically covers systems in which the operating system is mirrored using two physical disks. While not a strict requirement, adding extra disks for the sole use of the Solaris Live Upgrade software is an option. If additional disks are not available, the system disk and its mirror should be large enough to partition into two boot environments. If the disks are not large enough, the mirror must be broken, with the mirror disk used for the Solaris Live Upgrade operation. Organizations should weigh the risks of temporarily breaking the mirror against adding extra disks or updating to larger boot disks.

Preparing for a Live Upgrade

Before patching a system, it is important to properly mirror the boot environment and take care of any special directories.

Initial Disk Layout at Build Time

The Solaris Live Upgrade software makes it possible to apply patches to a mirrored system without breaking the mirror or disrupting user access. The procedures that

follow assume the system resides on disk `c1t0` and is mirrored onto disk `c1t1`. The recommended partitioning layout for both disks in the mirror is shown below.

```
/             size=10gb    cxtxdxs0   lvm=d0 sd=d10 and d20
swap          size=mem     cxtxdxs1   lvm=d1 sd=d11 and d21
/var          size=5gb     cxtxdxs3   lvm=d3 sd=d13 and d23
alt_lu_root   size=10gb    cxtxdxs4   lvm=d4 sd=d14 and d24
alt_lu_var    size=5gb     cxtxdxs5   lvm=d5 sd=s15 and d25
metadb        size=64k     cxtxdxs7    -c3
softpart      size=rest    cxtxdxs6   lvm=d6 sd=d16 and d26
```

The `/`, `swap`, and `/var` partitions belong to the system. The `alt_lu_root` and `alt_lu_var` partitions are copies of the `/` and `/var` partitions that can be patched while the system continues to run. The `softpart` partition consumes the remainder of the disk and provides the space in which all logical volumes and file systems can reside. As a result, additional file systems on the root disks should be created on device `d6`. For example:

```
# metainit d100 -p d6 10gb
```

Uprade to the Solaris Live Upgrade Software

Users should upgrade to the latest version of the Solaris Live Upgrade software prior to patching the system, regardless of the version of the Solaris OS running on the system. Solaris Live Upgrade is designed and compiled to work on the Solaris 8, 9, and 10 OS releases.

1. Install the latest Solaris Live Upgrade packages using the `liveupgrade20` script. The script runs silently and installs the latest Solaris Live Upgrade packages. To use a graphical version of the tool on the CDE or GNOME desktop environment, run the following command without the `-noconole` and `-nodisplay` options.

```
On SPARC systems:
# cd /cdrom/cdrom0/s0/Solaris_10/Tools/Installers
# ./liveupgrade20 -noconsole -nodisplay

On x86 and x64 systems:
# cd /cdrom/cdrom0/Solaris_10/Tools/Installers
# ./liveupgrade20 -noconsole -nodisplay
```

2. Install the latest Solaris Live Upgrade patches and other required patches as detailed in *Solaris Live Upgrade Software: Minimum Patch Requirements* on the Sun support site located at *sunsolve.sun.com* (Document ID: 72099). It is very important to apply the appropriate patches listed in this document prior to running any Solaris Live Upgrade commands.

Applying Patches with the Solaris Live Upgrade Software

Applying patches using the Solaris Live Upgrade software requires the creation of an alternate boot environment, and the application of baseline patches as well as any patches needed for advanced operating system features.

Creating the Alternate Boot Environment

The following steps explain how to create a copy of the boot environment that can be patched using the Solaris Live Upgrade software.

1. Create the alternate boot environment to be used during the Solaris Live Upgrade software patch process.

    ```
    # lucreate -C /dev/dsk/c1t0d0s0 -m /:/dev/md/dsk/d4:ufs \
    -m /var:/dev/md/dsk/d5:ufs -n newroot
    ```

2. Display the status of available boot environments using the `lustatus` command and verify that creation of the alternate boot enviornment is complete.

    ```
    # lustatus

    Boot Environment     Is        Active Active    Can    Copy
    Name                 Complete  Now    On Reboot Delete Status
    ---------------      --------  ------ --------- ------ ----------
    d0                   yes       yes    yes       no     -
    newroot              yes       no     no        yes    -
    ```

Using the Enterprise Installation Services Patch CD as the Baseline Patching Methodology

The following steps use the Enterprise Installation Services CD to baseline patches. However, the sample procedure can be used for any patch bundle, including patches recommended via the SunSolve℠ program, security and one off patches, custom patch bundles, or Sun℠ Update Connection enterprise patches. Some patches require the system to be rebooted before further patches can be applied. However, a system reboot is not necessary when using the Solaris Live Upgrade software to patch systems.

Solaris 10 Operating System

The following steps outline the procedure for applying patches to the Solaris 10 OS. For systems which place /opt on its own partition, refer to Appendix B for information on how to place /opt in a soft partition after the operating system build.

1. Change to the Solaris 10 OS patches directory on the Enterprise Installation Services CD.

    ```
    # cd EISCD~/patches/10
    ```

2. For one off patches, use the `unpack_patches` command and choose the default `/tmp` location. Next, use the `luupgrade` command to apply the patches placed in `/tmp` to the alternate boot environment. Note that the Enterprise Installation Services patch CD is set up in layers. The `patch_order` parameter is a file that contains the patch order for the directory after the `unpack-patches` command is run.

```
# luupgrade -n newroot -t -s /tmp/10/10_Recommended patch_order
# luupgrade -n newroot -t -s /tmp/10 patch_order
```

3. For distributed patching, use the `unpack_patches` command and choose the `/net/`*NFS_host*`/patchdir` location. Next, use the use the `luupgrade` command to apply the patches placed in `/net/`*NFS_host*`/patchdir` to the alternate boot environment.

```
# luupgrade -n newroot -t -s /net/NFS_host/patchdir/10/10_Recommended patch_order
# luupgrade -n newroot -t -s /net/NFS_host/patchdir/10 patch_order
```

Solaris 8 and 9 Operating System

Using the previously described `luupgrade` process on systems running the Solaris 8 and 9 OS can result in missing drivers and patches. For example, storage area network (SAN) modules such as `SUNWjfca` and linked driver patches can be involved in Solaris OS versions prior to the Solaris 10 OS. As a result, administrators should use the `patchadd` command to install recommended patch clusters and SAN patches in these environments.

1. Mount the inactive boot environment.

```
# lumount newroot
```

2. Change to the Solaris 8 or 9 OS patches directory on the Enterprise Installation Services CD.

```
# cd EISCD~/patches/8
OR
# cd EISCD~/patches/9
```

3. For one off patches, use the `unpack_patches` command and choose the default `/tmp` location. Next, use the `patchadd` command to apply the patches placed in `/tmp` to the alternate boot environment.

```
# patchadd -R /.alt.newroot -M  /tmp/8/8_Recommended patch_order
# patchadd -R /.alt.newroot -M  /tmp/8 patch_order
# patchadd -R /.alt.newroot -M  /tmp/SAN patch_order
```

4. For distributed patching, use the `unpack_patches` command and choose the `/net/`*NFS_host*`/patchdir` location. Next, use the use the `patchadd` command to apply the patches placed in `/net/`*NFS_host*`/patchdir` to the alternate boot environment.

```
# patchadd -R /.alt.newroot -M  /net/<NFS HOST>/patchdir/8/8_Recommended  patch_order
# patchadd -R /.alt.newroot -M  /net/<NFS HOST>/patchdir/8 patch_order
# patchadd -R /.alt.newroot -M  /net/<NFS HOST>/patchdir/SAN patch_order
```

5. Unmount the inactive boot environment.

```
# luunmount newroot
```

Patching Systems Running the Solaris 10 OS and Solaris Containers

The Solaris 10 OS 8/07 contains Solaris Live Upgrade support for patching and updating systems which use Solaris Containers and Solaris Zones technology. The Solaris Live Upgrade components that are part of the Solaris 10 8/07 OS can be used on earlier releases of the Solaris 10 OS to enable the patching of Solaris Zones using the Solaris Live Upgrade software. For example, a system running the Solaris 10 11/06 OS with Solaris Zones can be patched using the standard Solaris Live Upgrade software. To do so, obtain the latest Solaris Live Upgrade packages from the latest Solaris OS release, use the `liveupgrade20` script, and patch the system prior to running the Solaris Live Upgrade process per the recommendations outlined in *Solaris Live Upgrade Software: Minimum Patch Requirements* on the Sun support site located at *sunsolve.sun.com* (Document ID: 72099).

Note – Sun intends to update this Sun BluePrints article to show examples of patching with Solaris Zones after the release of the Solaris 10 8/07 OS. Use of the Solaris Live Upgrade software with Solaris Zones is not supported without the 8/07 release or later packages installed.

Booting the Patched System

In order to run the patched system, the machine must be rebooted. Prior to rebooting, set the new reboot to boot off the alternative environment `newroot`. This step changes the boot device to be the `newroot` partition at the next shutdown.

```
# luactivate newroot
```

Note – Either the `/sbin/init` or `/usr/sbin/shutdown` command ***must*** be used to shutdown the system prior to the reboot process.

Note – When the `luactivate` command is run, the system performs a simple synchronization to identify changes made between the running of the `luactivate` and `luupgrade` commands. For example, changes may occur to the `passwd` file. More information can be found in the `synclist(4) man` page.

Subsequent Patching

The example in this document describes the initial Solaris Live Upgrade software setup. Configuring the system using these procedures results in two bootable environments — the original and patched boot environments. Administrators can boot off the last patched system and use the other copy as a pre-patch, bootable mirrored backup.

The original environment should not be deleted until it is time to perform the next patch cycle, or the new environment has run for a sufficient amount of time to ensure any problems have surfaced. Following these guidelines enables a quick rollback to the previous version, if required.

After the host has run on the new patched operating system for a sufficient time, performing a Solaris Live Upgrade on the original boot environment to which new patches have not been applied is recommended. This provides an alternate backup boot environment in the event operating system issues arise between patching cycles. At the next patch cycle, delete the upgraded backup environment now located on device `d0` and re-run the Solaris Live Upgrade. Doing so incorporates changes that may have taken place between patch cycles. The backup alternate environment will not be active unless the `luactivate` command is run or the boot device is changed.

```
# ludelete d0

# lucreate -C /dev/dsk/c1t1d0s4 -m /:/dev/md/dsk/d0:ufs -m \
/var:/dev/md/dsk/d3:ufs -n d0
```

Summary

Operating system patching is a natural part of the platform lifecycle. Unfortunately, traditional patching methods require systems to be taken offline for the duration of the patching process. By creating a copy of the boot environment and patching it with the Solaris Live Upgrade software, existing systems can continue to run throughout the patching process. With these tools and procedures, administrators are better able to create updated operating system environments without disrupting users with the security of knowing the previous version can be booted at any time.

About the Author

Since joining Sun in 1995, Jeff Smith has worked as a system support engineer helping create robust high-end server deployment environments. Today, Jeff works in the Sun Solutions Center solving customer problems. Prior to joining Sun, Jeff worked at Apollo Computer, Inc., Hewlett-Packard, and Fujitsu.

Acknowledgements

The author would like to recognize the following individuals for their contributions to this article:

- Rodney Linder and Tong-Pheng Koh of Sun Microsystems, for performing the initial investigation into the techniques described in this document

References

"Configuring Solaris Live Upgrade 2.0, Command Line"
To access this document online, visit the Sun support site located at
`sunsolve.sun.com` and search for document ID 43596.

Ordering Sun Documents

The SunDocs℠ program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is
`http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at:
`http://www.sun.com/blueprints/online.html`

Appendix A

Example Output from Patching the Solaris 9 OS

```
# lucreate –C /dev/dsk/c1t1d0s0  –m /:/dev/md/dsk/d4:ufs –m /var:/dev/md/dsk/d5:ufs –n newroot

Discovering physical storage devices
Discovering logical storage devices
Cross referencing storage devices with boot environment configurations
Determining types of file systems supported
Validating file system requests
Preparing logical storage devices
Preparing physical storage devices
Configuring physical storage devices
Configuring logical storage devices
Analyzing system configuration.
No name for current boot environment.
INFORMATION: The current boot environment is not named - assigning name <d0>.
Current boot environment is named <d0>.
Creating initial configuration for primary boot environment <d0>.
The device </dev/dsk/c1t1d0s0> is not a root device for any boot environment.
PBE configuration successful: PBE name <d0> PBE Boot Device </dev/dsk/c1t1d0s0>.
Comparing source boot environment <d0> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Searching /dev for possible boot environment filesystem devices

Updating system configuration files.
The device </dev/dsk/c1t1d0s4> is not a root device for any boot environment.
Creating configuration for boot environment <newroot>.
Creating boot environment <newroot>.
Creating file systems on boot environment <newroot>.
Creating <ufs> file system for </> on </dev/md/dsk/d4>.
Creating <ufs> file system for </var> on </dev/md/dsk/d5>.
Mounting file systems for boot environment <newroot>.
Calculating required sizes of file systems for boot environment <newroot>.
Populating file systems on boot environment <newroot>.
Checking selection integrity.
Integrity check OK.
Populating contents of mount point </>.
Populating contents of mount point </var>.
Copying.
Creating shared file system mount points.
Creating compare databases for boot environment <newroot>.
Creating compare database for file system </var>.
Creating compare database for file system </>.
Updating compare databases on boot environment <newroot>.
Making boot environment <newroot> bootable.
Setting root slice to Solaris Volume Manager metadevice </dev/md/dsk/d4>.
Population of boot environment <newroot> successful.
Creation of boot environment <newroot> successful.
```

```
# lustatus
Boot Environment            Is       Active Active    Can    Copy
Name                        Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
d0                          yes      yes    yes       no     -
newroot                     yes      no     no        yes    -

# lumount newroot

# df -h

/dev/md/dsk/d4         10G   1.4G   8.6G    31%    /.alt.newroot
/dev/md/dsk/d5         5G    29M    4.9G    2%     /.alt.newroot/var

# pwd
/tmp/9/9_Recommended

# patchadd -R /.alt.newroot -M /tmp/9/9_Recommended ./patch_order


Checking installed patches...
Verifying sufficient filesystem capacity (dry run method)...
Installing patch packages...
...
Patch number 116340-05 has been successfully installed.
See /.alt.newroot/var/sadm/patch/116340-05/log for details.

Patch packages installed:
SUNWgzip
SUNWsfinf

Checking installed patches...
Verifying sufficient filesystem capacity (dry run method)...


...

# ls -l /dev/rdsk/c1*2
lrwxrwxrwx   1 root     root           47 Nov 23 13:12 /dev/rdsk/c1t1d0s2
-> ../../devices/pci@1f,4000/scsi@3,1/sd@1,0:c,raw

lrwxrwxrwx   1 root     root           47 Nov 23 13:12 /dev/rdsk/c1t2d0s2
-> ../../devices/pci@1f,4000/scsi@3,1/sd@2,0:c,raw

# luactivate newroot


********************************************************************

The target boot environment has been activated. It will be used when you
reboot. NOTE: You MUST NOT USE the reboot, halt, or uadmin commands. You
MUST USE either the init or the shutdown command when you reboot. If you
do not use either init or shutdown, the system will not boot using the
target BE.

********************************************************************
```

```
In the event a failure occurs when booting the target boot environment on
SPARC® platforms, the following process must be followed to fallback to
the currently working boot environment.At the PROM monitor (ok prompt)

To boot from a Solaris OS CD:
ok> boot cdrom -s

To boot from the network:
ok> boot net -s

Mount the current boot environment root slice to a directory, such as /
mnt, using the following mount command.
# mount -Fufs /dev/dsk/c1t1d0s0 /mnt

Run the luactivate utility without any arguments from the current boot
environment root partition.
# /mnt/sbin/luactivate

The luactivate command activates the previous working boot environment
and indicates the result.

Exit single user mode and reboot the system.

Activation of boot environment newroot successful
```

```
# lustatus

Boot Environment     Is            Active  Active     Can     Copy
Name                 Complete      Now     On Reboot  Delete  Status
-------------------- --------      ------  ---------  ------  ----------
d0                   yes           yes     no         no      -
newroot              yes           no      yes        no      -

# eeprom |grep boot-device   ( BEFORE REBOOT )
boot-device=/pci@1f,4000/scsi@3,1/disk@1,0:a  /pci@1f,4000/scsi@3/
disk@1,0:a

# init 6

Come back off the new boot environment.

# df -h
/dev/md/dsk/d4        10G     1.4G    8.6G    31%    /
/dev/md/dsk/d5        5G      29M     4.9G    2%     /var

# lustatus
Boot Environment     Is            Active  Active     Can     Copy
Name                 Complete      Now     On Reboot  Delete  Status
-------------------- --------      ------  ---------  ------  ----------
d0                   yes           yes     no         no      -
newroot              yes           no      yes        no      -

# eeprom |grep boot-device   ( AFTER REBOOT )
boot-device=/pci@1f,4000/scsi@3,1/disk@1,0:e /pci@1f,4000/scsi@3/
disk@1,0:a

# lustatus
Boot Environment     Is            Active  Active     Can     Copy
Name                 Complete      Now     On Reboot  Delete  Status
-------------------- --------      ------  ---------  ------  ----------
d0                   yes           no      no         yes     -
newroot              yes           yes     yes        no      -
```

Appendix B

Example of /opt on Root as an Extra Soft Partition

In the Solaris OS, the disk vtoc is limited to seven available partitions. If a directory such as /opt needs to reside on its own partition, it can be placed in a soft partition after the operating system build. The example steps below assume /opt resides on soft partition device d61 which is part of the d6 mirrored partition.

1. Create soft partitions for the /opt and /alt_lu_opt file systems.

    ```
    # metainit –d61 –p d6 5gb
    # metainit –d62 –p d6 5gb
    ```

2. Create a new file system on device d61 using the newfs command.

    ```
    # newfs /dev/md/dsk/d61
    ```

3. Mount the newly created file system onto the /opt1 directory.

    ```
    # mount –t ufs /dev/md/dsk/d61 /opt1
    ```

4. Copy the contents of the /opt directory to the /opt1 directory.

5. Shut the system down to single user mode.

    ```
    # halt
    ...
    # boot -s
    ```

6. Save the /opt directory as the /opt.old directory.

    ```
    # mv /opt /opt.old
    ```

7. Mount the contents of /opt1, which resides on device d61, onto the /opt directory.

    ```
    # mount –t ufs /dev/md/dsk/d61 /opt
    ```

8. Verify that device d61 is mounted onto the /opt directory using the df command.

```
# df -k

/dev/md/dsk/d4          4.8G    1.4G    3.3G    31%    /
/proc                    0K      0K      0K     0%    /proc
mnttab                   0K      0K      0K     0%    /etc/mnttab
fd                       0K      0K      0K     0%    /dev/fd
/dev/md/dsk/d5          1.9G     34M    1.8G     2%    /var
swap                    1.2G     40K    1.2G     1%    /var/run
swap                    1.2G     72K    1.2G     1%    /tmp
/dev/md/dsk/d61         963M    1.5M    904M     1%    /opt
```

9. Print status information for the devices on the system and verify mirrors.

```
# metastat -p
d5 -m d15 d25 1
d15 1 1 c1t1d0s5
d25 1 1 c1t2d0s5
d4 -m d14 d24 1
d14 1 1 c1t1d0s4
d24 1 1 c1t2d0s4
d3 -m d13 d23 1
d13 1 1 c1t1d0s3
d23 1 1 c1t2d0s3
d1 -m d11 d21 1
d11 1 1 c1t1d0s1
d21 1 1 c1t2d0s1
d0 -m d10 d20 1
d10 1 1 c1t1d0s0
d20 1 1 c1t2d0s0
d62 -p d6 -o 2097154 -b 2097152
d6 -m d16 d26 1
d16 1 1 c1t1d0s6
d26 1 1 c1t2d0s6
d61 -p d6 -o 1 -b 2097152
```

10. Update the vfstab file to include the new /opt file system for future reboots.

```
/dev/md/dsk/d61 /dev/md/rdsk/d61 /opt ufs 2 yes logging
```

11. Create the alternate boot environment to be used during the Solaris Live Upgrade patch process.

```
# lucreate -C /dev/dsk/c1t0d0s0 -m /:/dev/md/dsk/d4:ufs \
-m /var:/dev/md/dsk/d5:ufs -m /opt:/dev/md/dsk/d62:ufs -n newroot
```