開
放
的
열린
مفتوح
libre
मुक्त
ముక్త
livre
libero
ಮುಕ್ತ
开放的
açık
open
nyílt
⠠⠠⠠⠠
פתוח
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
வெளிப்படை

Live Upgrade: Using and Abusing

- Boyd Adamson

What is the problem?

Upgrades and Patches are the problem

- Outage time – even some patches take ages
- Risk – what happens if it all goes wrong?

What are admins doing now?

- Backups
- Break a root mirror, upgrade half, fall back on the other half.

But neither of these solves the outage problem.

How do we address this?

- The break-a-mirror approach has the right idea
- Provides fall-back
- Doesn't solve the outage time issue

...

How do we address this?

- The break-a-mirror approach has the right idea
- Provides fall-back
- Doesn't solve the outage time issue

So...

- Use the same idea (make a copy)
- Add more flexibility
- Teach the patch and upgrade tools about it

How does this work?

Define a thing called a “boot environment” (BE).

- Basically, it's `/`, `/usr`, `/var`
- Maybe `swap`
- Maybe other filesystems (like `/opt`, `/export/home`)

How does this work?

- You create a new (alternate) BE
  - Normally copy the running OS
  - Can be populated with a Flash Archive
- Upgrade or patch the new BE
  - This is live. The OS is still running in the “old” BE
- Reboot into new BE

...

How does this work?

- You create a new (alternate) BE
  - Normally copy the running OS
  - Can be populated with a Flash Archive
- Upgrade or patch the new BE
  - This is live. The OS is still running in the “old” BE
- Reboot into new BE

Total outage: Time for reboot
(+ manifest import)

Fall-back option: boot back into old BE

Basic operation

Creation:

- Make sure the partitions you need exist first
- They don't need to be the same size

`lucreate`

`-c` *current_be_name* — For first time creation only

`-n` *new_be_name*

`-m` *mountpoint:device:options* — Repeat as needed

Creation Example

- We start with only `/` and `swap`.
- Make a new BE on a second disk. Preserve partition layout

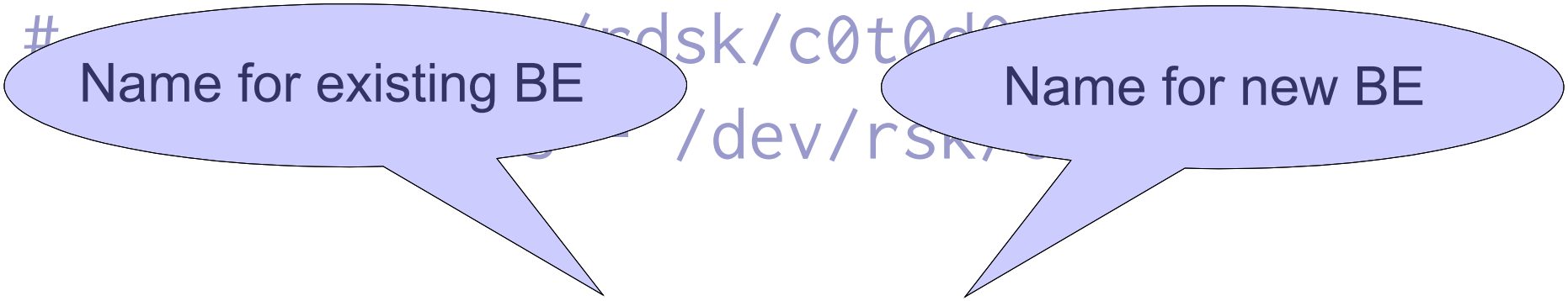
Creation Example

- We start with only / and swap.
- Make a new BE on a second disk. Preserve partition layout:

```
# prtvtoc /dev/rdsk/c0t0d0s2 |
    fmthard -s - /dev/rsk/c0t8d0s2
```

Creation Example

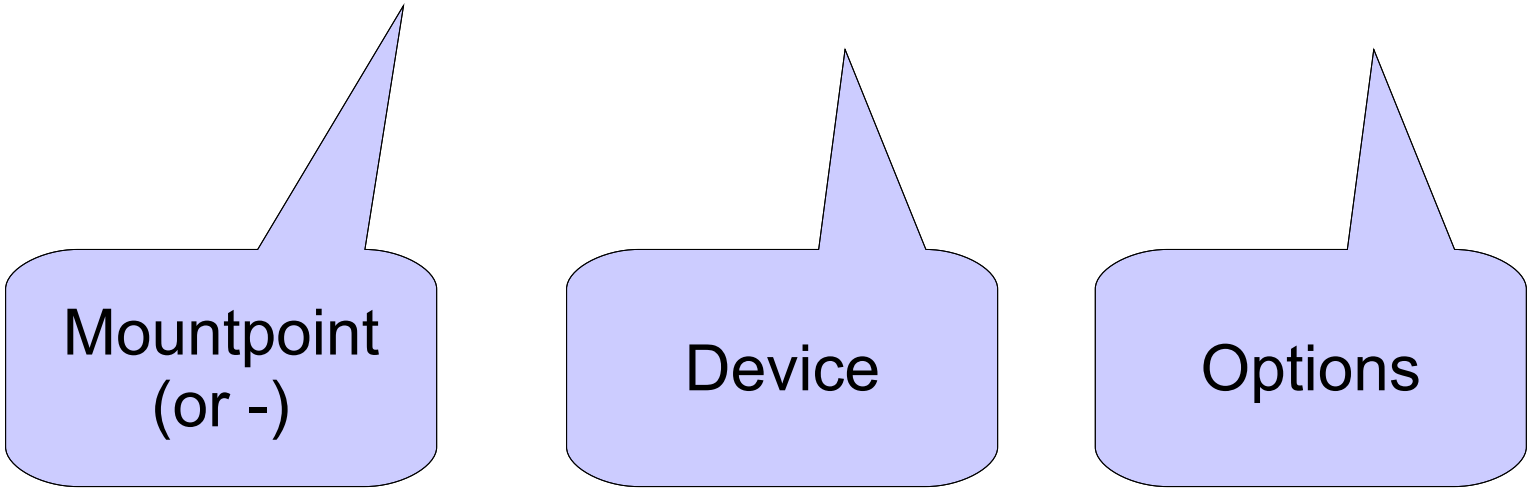
- We start with only / and swap.
- Make a new BE on a second disk. Preserve partition layout:




```
# lucreate -c disk0 -n disk1
    -m /:/dev/dsk/c0t8d0s0:ufs
    -m -:/dev/dsk/c0t8d0s1:swap
```

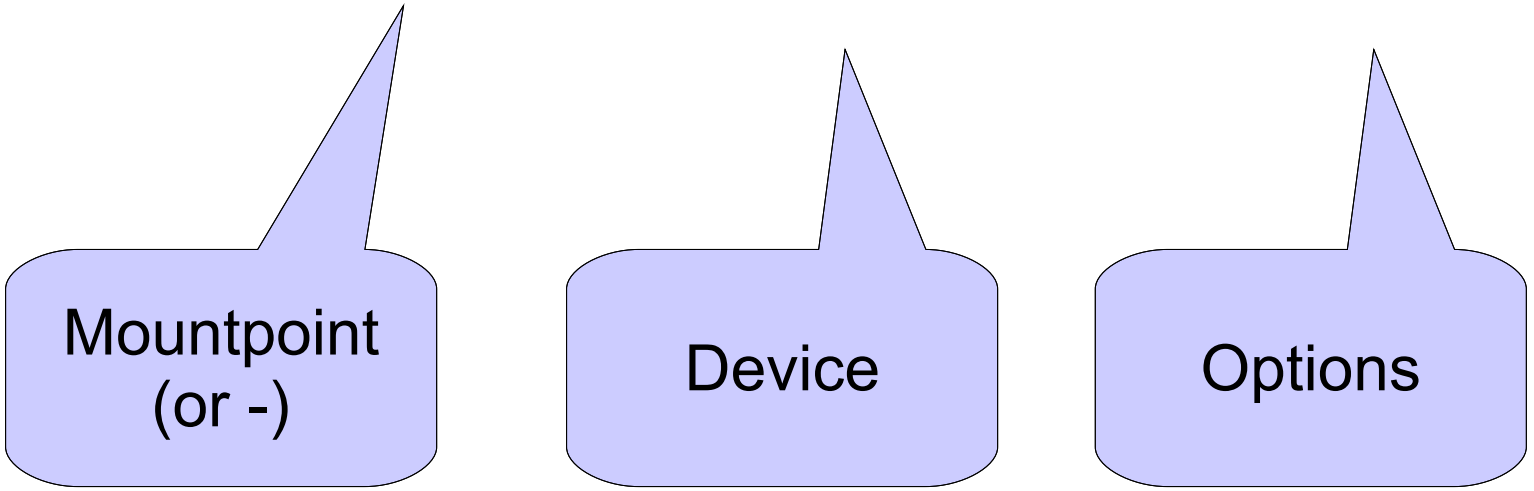
Filesystem specification

```
-m /:/dev/dsk/c0t8d0s0:ufs
```



Another example

```
-m -:/dev/dsk/c0t8d0s1:swap
```



So, what do we get?

```
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
disk0                      yes      yes    yes       no     -
disk1                      yes      no     no        yes    -

# lufslist -n disk1
               boot environment name: disk1

Filesystem              fstype    device size Mounted on          Mount Options
----------------------- -------- ------------ ------------------- --------------
/dev/dsk/c0t8d0s1       swap       1075576320 -                   -
/dev/dsk/c0t8d0s0       ufs        7955587584 /                   -
```

What about other filesystems?

- Other filesystems (e.g. `/opt`, `/export/home`) are *shared* unless you mention them
- This happens with `swap` too
- If you *do* mention them with `-m` they will be copied

Changing filesystem layout

- The new BE can have a different filesystem layout to the old.
- For example you can split `/` and `/usr`
- Or you can merge them
- Some people use LU just for this feature

Filesystem Split Example

Before:

```
# lufslist -n disk0
               boot environment name: disk0
               This boot environment is currently active.
               This boot environment will be active on next system boot.

Filesystem              fstype    device size Mounted on          Mount Options
----------------------- -------- ------------ ------------------- --------------
/dev/dsk/c0t0d0s1       swap       1075576320 -                   -
/dev/dsk/c0t0d0s0       ufs        7955587584 /                   -
```

Filesystem Split Example

Just name `/usr` explicitly and it will be split from `/`

```
# lucreate -n disk1
  -m /:/dev/dsk/c0t8d0s0:ufs
  -m -:/dev/dsk/c0t8d0s1:swap
  -m /usr:/dev/dsk/c0t8d0s3:ufs
```

Filesystem Split Example

After:

```
# lufslist -n disk1
               boot environment name: disk1

Filesystem              fstype    device size Mounted on          Mount Options
----------------------- -------- ------------ ------------------- --------------
/dev/dsk/c0t8d0s1       swap       1075576320 -                   -
/dev/dsk/c0t8d0s0       ufs        1075576320 /                   -
/dev/dsk/c0t8d0s3       ufs        6880011264 /usr                -
```

Filesystem Merge

- Since `/usr` and `/var` cannot be shared, if you don't mention them they will be merged into `/`
- For other filesystems:

```
-m /export/home:merged:ufs
```

- This will merge (by copying) `/export/home` into it's parent (probably `/`)

What if I have a mirrored boot disk?

- With SVM, LU can break a mirror and preserve the contents
- It will create a new mirror for the new BE
- It can also create a mirror for the new BE when you don't already have one

Mirrored root example

Before:

```
# df -h /
Filesystem             size   used  avail capacity  Mounted on
/dev/md/dsk/d10        7.3G   3.8G   3.5G    53%    /
# metastat -c
d10                  m  7.4GB d11 d12
    d11              s  7.4GB c0t0d0s0
    d12              s  7.4GB c0t8d0s0
```

Mirrored Root example

```
# lucreate -n disk1
  -m /:/dev/md/dsk/d20:ufs,mirror
  -m /:/dev/dsk/c0t8d0s0:detach,attach,preserve

...

Preserving <ufs> file system for </> on
  </dev/md/dsk/d20>.
```

Mirrored root example

After:

```
# metastat -c
d20                m  7.4GB d12
    d12            s  7.4GB c0t8d0s0
d10                m  7.4GB d11
    d11            s  7.4GB c0t0d0s0
```

Ok, so what do I do with it?

Once you have an alternate BE (ABE) you can make changes to it

- Patch
- Upgrade
- Add and remove packages
- Just edit

Upgrades, Patches, and Packages

- luupgrade is your friend
- It does a lot

. . .

Upgrades, Patches, and Packages

- luupgrade is your friend
- It does a lot

```
# luupgrade

ERROR: At least one option from <-c, -C, -f, -i, -I, -p, -P, -t, -T, -u> must be specified.

USAGE: luupgrade [ -u | -f | -p | -r | -P | -R | -i | -c ] [ -l error_log ] [ -o outfile ] [ -N ] [ -X ] [ additional optional and
     required parameters ]:

OS Upgrade:      luupgrade -u -n BE_name [ -l error_log ] [ -o outfile ] [ -N ] [ -X ] [ -D ] -s source_os_image_path [ -j profile_path ]

Flash Upgrade:   luupgrade -f -n BE_name [ -l error_log ] [ -o outfile ] [ -N ] [ -X ] [ -D ] -s source_os_image_path ( -a archives | -j
     profile_path | -J profile )

Add Packages:    luupgrade -p -n BE_name [ -l error_log ] [ -o outfile ] [ -N ] [ -X ] ( ( -s|-d ) source_packages_path ) [ -a
     pkg_admin_file ] [ -O pkgadd_options ] [ pkginst [ pkginst... ] ]

Remove Packages: luupgrade -P -n BE_name [ -l error_log ] [ -o outfile ] [ -N ] [ -X ] [ -O pkgadd_options ] pkginst [ pkginst... ]

Check Packages:  luupgrade -C -n BE_name [ -l error_log ] [ -o outfile ] [ -N ] [ -X ] [ -O pkgchk_options ] [ pkginst [ pkginst... ] ]

Package Info:    luupgrade -I -n BE_name [ -l error_log ] [ -o outfile ] [ -N ] [ -X ] [ -O pkginfo_options ] [ pkginst [ pkginst... ] ]

Add Patches:     luupgrade -t -n BE_name [ -l error_log ] [ -o outfile ] [ -N ] [ -X ] -s source_patches_path [ -O patchadd_options ]
     [ patchname [ patchname... ] ]

Remove Patches:  luupgrade -T -n BE_name [ -l error_log ] [ -o outfile ] [ -N ] [ -X ] [ -O patchrm_options ] patchname [ patchname... ]

Run Installer:   luupgrade -i -n BE_name [ -l error_log ] [ -o outfile ] [ -N ] [ -X ] -s install_image_path [ -O installer_options ]

Check Media:     luupgrade -c [ -l error_log ] [ -o outfile ] [ -X ] -s image_path

INFORMATION: Any BE_name or options should be enclosed in single quotes.
```

Patching

- Need the patches already unpacked
- Specify which patches to install
  - patch_order file
  - directory and list of patches
  - directory and '*' (Solaris 10 and later – will work out the order)

```
# luupgrade -t -n disk1 -s /var/tmp/patches '*'
```

Package manipulation

Just calls `pkgadd/pkgrm` (with `-R`)

Add:

```
# luupgrade -p -n disk1
    -s /export/packages BOYDzsh
```

Remove:

```
# luupgrade -P -n disk1 BOYDzsh
```

Upgrade the OS

Wait! The Solaris 9 LU tools don't know how to upgrade to Solaris 10

- In general, install the “destination” version LU packages first.
- Make sure the machine is patched up to date before doing an upgrade.
  - See SunSolve Infodoc 72099 for minimum patch requirements

Installing the LU packages

```
# cd /mnt/Solaris_10/Tools/Installers
# ./liveupgrade20
```

or

```
# pkgrm SUNWlucfg SUNWlur SUNWluu
# cd /mnt/Solaris_10/Product
# pkgadd -d . SUNWlucfg SUNWlur SUNWluu
```

Note: `SUNWlucfg` only exists in s10u4 and Nevada (from b53)

Perform the upgrade

```
# luupgrade -u -n disk1 -s /net/drain/export/install/nv_64_sparc
...
Performing the operating system upgrade of the BE <disk1>.
...
INFORMATION: The file </var/sadm/system/logs/upgrade_log> on boot
environment <disk1> contains a log of the upgrade operation.
INFORMATION: The file </var/sadm/system/data/upgrade_cleanup> on boot
environment <disk1> contains a log of cleanup operations required.
INFORMATION: Review the files listed above. Remember that all of the
   files
are located on boot environment <disk1>. Before you activate boot
environment <disk1>, determine if any additional system maintenance is
required or if additional media of the software distribution must be
installed.
The Solaris upgrade of the boot environment <disk1> is complete.
```

Other changes

- You can mount other BEs and manipulate their contents

```
# lumount disk1 /mnt
```

- This mounts all the filesystems for the BE under `/mnt`
- You can edit files then:

```
# luumount disk1
```

- Note, you don't have to worry about editing `/etc/vfstab` in most cases

Using the new BE

When you're ready to use the new BE you "activate" it:

```
# luactivate disk1
**********************************************************************

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.

**********************************************************************
...
Activation of boot environment <disk1> successful.
```

Using the new BE

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

In case of a failure while booting to the target BE, the following process
needs to be followed to fallback to the currently working boot environment:

1. Enter the PROM monitor (ok prompt).

2. Change the boot device back to the original boot environment by typing:

     setenv boot-device /pci@1f,4000/scsi@3/disk@0,0:a

3. Boot to the original boot environment by typing:

     boot

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

After luactivate

```
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
disk0                      yes      yes    no        no     -
disk1                      yes      no     yes       no     -
```

- The real work is done by shutdown scripts
- You *must* use `shutdown` or `init`, not `reboot` or `halt`

What exactly does luactivate do?

The shutdown scripts:

- Change eeprom settings on SPARC
- Change the GRUB menu on x86:

```
Solaris 10 11/06 s10x_u3wos_10 X86
Solaris failsafe
disk1
disk1 failsafe
disk0
disk0 failsafe
```

After luactivate

```
# init 6
...
SunOS Release 5.11 Version snv_64a 64-bit
Copyright 1983-2007 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
```

What about changed files?

- Files can change in the original BE after we create but before we activate the new BE
- On the *first* boot of a BE, data is copied from the source BE to bring the new BE up to date
- The list to copy is in `/etc/lu/synclist`. e.g:

```
/etc/default/passwd          OVERWRITE
/etc/dfs                     OVERWRITE
/var/log/syslog              APPEND
/var/adm/messages            APPEND
```

Limitations

- No live upgrade of systems with non-global zones until ZULU (Nevada b53 and Solaris 10 8/07)
  - Upgrades of systems with NG zones takes a *long* time, so LU is even more important
- Doesn't understand ZFS.
  - Can't upgrade Zones on ZFS (waiting on ZFS boot installer integration)
  - Mistakenly creates all zfs mountpoints on target BE, even those that should be on ZFS

More on the ZFS mountpoints

- Say you have `/export/home` and `/export/home/boyd` as ZFS filesystems
- LU will create `/export/home` *and* `/export/home/boyd` mountpoints in the new BE (should just make `/export/home`)

```
# svcs -x
svc:/system/filesystem/local:default (local file system mounts)
 State: maintenance since Mon Sep 10 07:04:32 2007
Reason: Start method exited with $SMF_EXIT_ERR_FATAL.
   See: http://sun.com/msg/SMF-8000-KS
   See: /var/svc/log/system-filesystem-local:default.log
Impact: 28 dependent services are not running.  (Use -v for list.)
```

More on the ZFS mountpoints

- Avoidance:

  Before `luactivate` & `reboot`: `lumount` then remove the extra mountpoints

- If you forgot:

  ```
  # zfs umount -a
  ```

  remove mountpoints

  ```
  # svcadm clear filesystem/local
  ```

Other gotchas

```
# lucreate -n disk1 -m /:/dev/dsk/c0t8d0s0:ufs
   -m /usr:/dev/dsk/c0t8d0s3:ufs
...
Template entry /usr:/dev/dsk/c0t8d0s3:ufs skipped
luconfig: ERROR: Template filesystem definition failed for
  /usr, all devices are not applicable..
ERROR: Configuration of boot environment failed.
```

Why!?

...

Other gotchas

```
# lucreate -n disk1 -m /:/dev/dsk/c0t8d0s0:ufs
   -m /usr:/dev/dsk/c0t8d0s3:ufs
...
Template entry /usr:/dev/dsk/c0t8d0s3:ufs skipped
luconfig: ERROR: Template filesystem definition failed for
  /usr, all devices are not applicable..
ERROR: Configuration of boot environment failed.
```

Why!?

```
Part      Tag    Flag     Cylinders        Size
  3 unassigned    wu    1170 - 4923        6.43GB
```

Yes, it cares. Flags must be “wm”. No, I don't know why.

Other gotchas

- (On x86)

```
# lucreate ...

...
ERROR: No matching BIOS id found for: </dev/dsk/c2d0s0>
ERROR: Cannot determine GRUB id for ABE disk </dev/dsk/c2d0s0>
ERROR: Unable to determine the configuration of the target boot environment
   <disk1>.
ERROR: Update of loader failed.
ERROR: Unable to umount ABE <disk1>: cannot make ABE bootable.
Making the ABE <disk1> bootable FAILED.
ERROR: Unable to make boot environment <disk1> bootable.
ERROR: Unable to populate file systems on boot environment <disk1>.
ERROR: Cannot make file systems for boot environment <disk1>.
```

Other gotchas

- Why?
- LU needs to get the BIOS id for disks to update the GRUB menu
- Some systems (like this U20) don't seem to provide it:

```
/sbin/biosdev
biosdev: Could not match any!!
```

- Should look like this:

```
# /sbin/biosdev
0x80 /pci@0,0/pci-ide@7/ide@0/cmdk@0,0
0x81 /pci@0,0/pci-ide@7/ide@1/cmdk@0,0
```

Other gotchas

- This is very like 6379907
- My “solution” (look away if you're squeamish):

...

Other gotchas

- This is very like 6379907
- My “solution” (look away if you're squeamish):

```
# mv /sbin/biosdev /sbin/biosdev.orig
# cat > /sbin/biosdev
cat << EOF
0x80 /pci@0,0/pci-ide@7/ide@0/cmdk@0,0
0x81 /pci@0,0/pci-ide@7/ide@1/cmdk@0,0
EOF
# chmod +x /sbin/biosdev
```

Recommendations

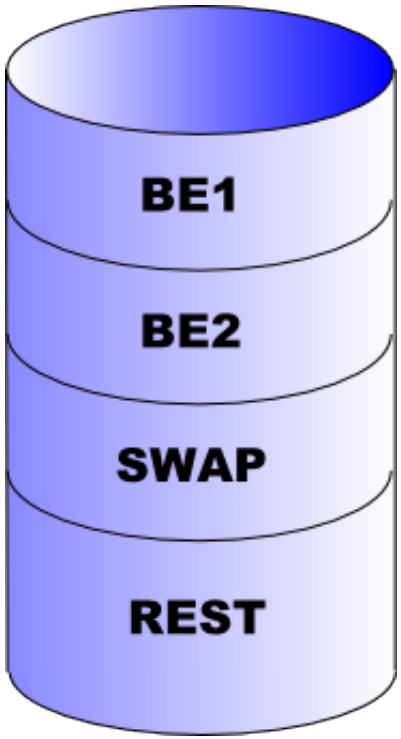
- LU is now the recommended way to do all patching
- ... and Upgrades

So...

- *Plan* your disk slices accordingly

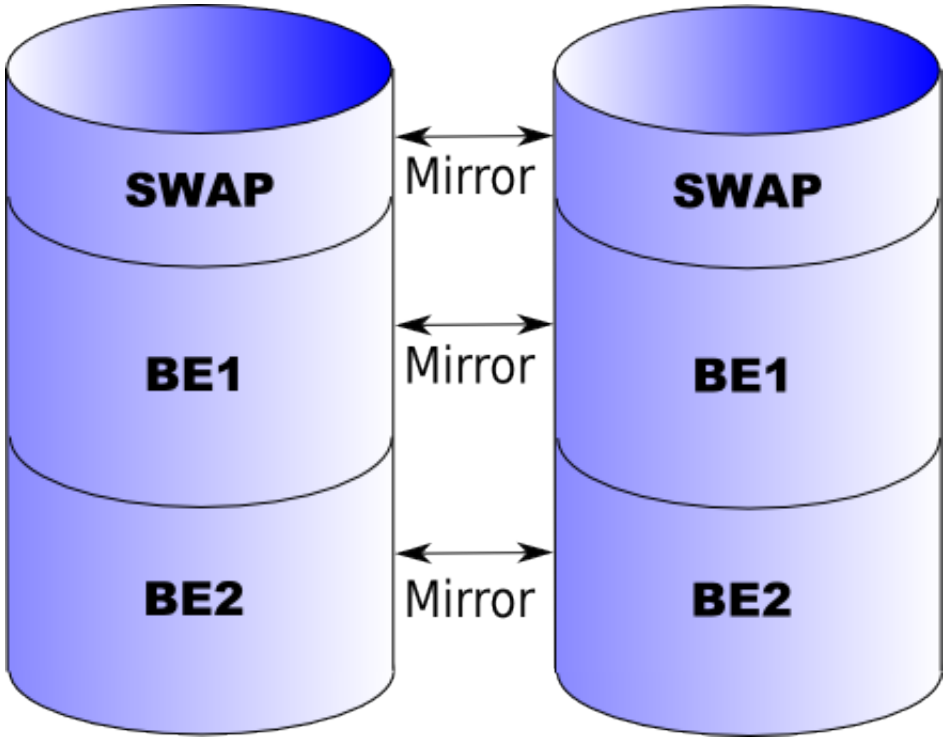
Example layouts

Single-disk machine:



Example layouts

Server. This is OS only. Data is on other disks (under ZFS of course):



Random other stuff

- If you're alternating between BEs, you don't have to `ludelete` and re-`lucreate` them all the time.
- `lumake` will re-populate an existing BE from the current one:

```
# lumake -n other_be
```

Random other stuff

More and different BEs:

- You can, of course have more than 2 BEs
- `lucreate -s be_name` populates from the named BE
- `lucreate -s -` leaves the BE empty
  - You can then use `luupgrade` to populate it with a Flash Archive

Random other stuff

- You can specify multiple destinations for each filesystem:

```
lucreate -m /:/dev/dsk/c0t4d0s0:ufs
         -m /:/dev/dsk/c0t4d0s1:ufs
         -m /:/dev/dsk/c0t4d0s5:ufs ...
```

- LU will use the first one that's not already part of a BE
- Makes scripts for alternating BEs easier

Random other stuff

- If you have a lot of filesystems that you use frequently, put them in a file and point lucreate at it:

```
# lucreate -n disk1 -M /some/file
```

The future

- The most annoying thing about LU is that we have to wait for the copying and have space for it.
- If only there were some way to clone the OS install in a space-efficient way...
- Everyone is waiting for ZFS boot :)

Any Questions?

開
放
的
열린
مفتوح
libre
मुक्त
ముక్త
livre
libero
ಮುಕ್ತ
开放的
açık
open
nyílt
⠂⠂⠂⠂⠂
פתוח
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
வெளிப்படை

Thank you!

Boyd Adamson

boyd-adamson@usa.net
http://quidocet.blogspot.com

“open” artwork and icons by chandan:
http://blogs.sun.com/chandan