



# LiveMedia Technologies for OpenSolaris

Moinak Ghosh  
Solaris Sustaining Engineering

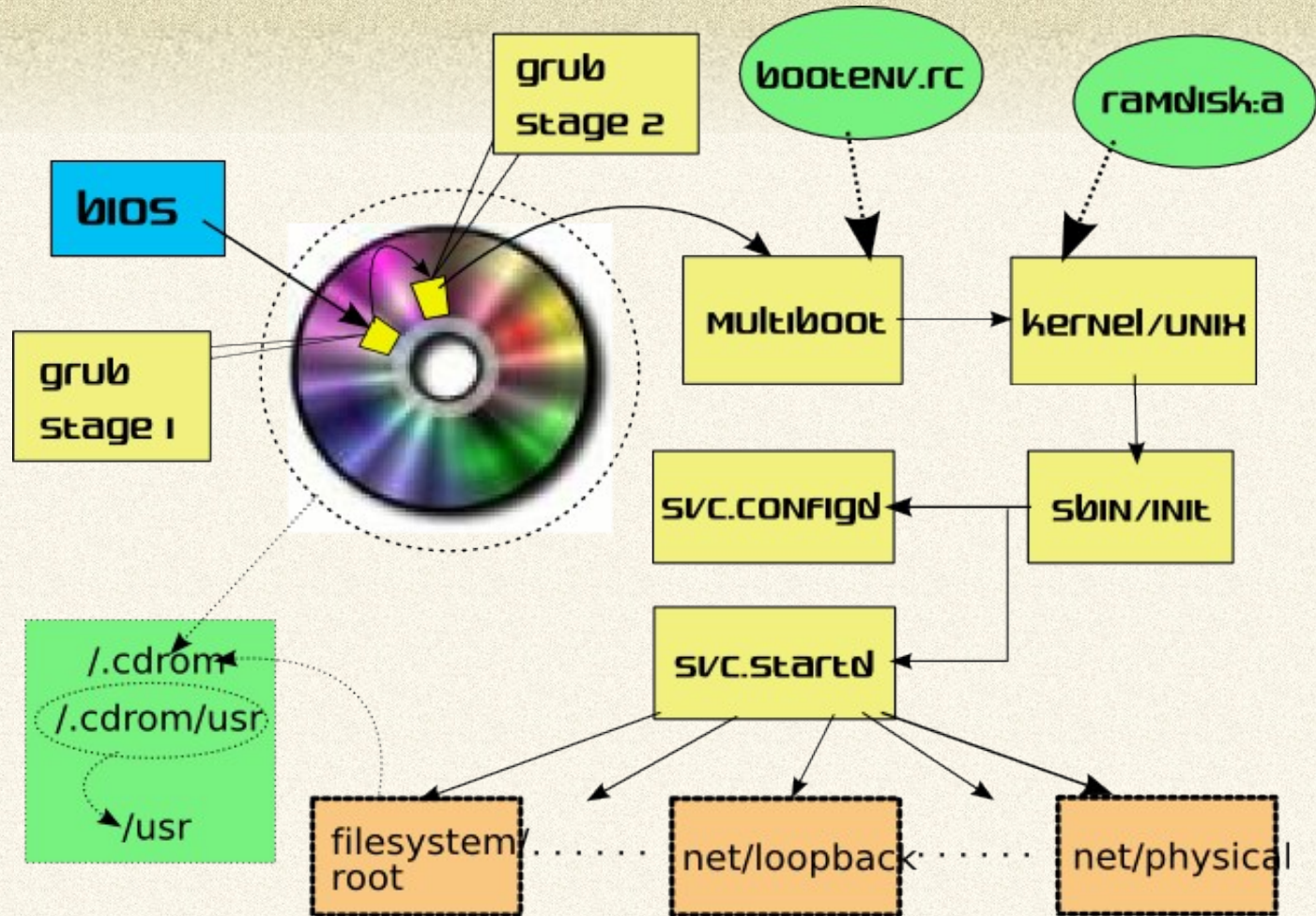
# Why LiveMedia ?

- Bootable removable media reduces barrier to entry
- Easy to test drive – no partitioning, installation
- Reaches out to a wider audience - visibility
- Good for System Recovery
- Good for Hardware Compatibility Testing
- Ideal for Install Media
- Enabler of Ubiquitous Computing (via writable Live Media).

# The Basic Mechanism

- The root filesystem resides in a RAM-resident segment – Ramdisk.
- Bootloader (GRUB Eltorito) loads basic Ramdisk and initial kernel
- The kernel initializes completely from Ramdisk
- Init and bare minimum system libs also in Ramdisk
- A startup service probes removable media
- /usr and other filesystems mounted directly from removable media.

# OpenSolaris LiveCD Bootup



Initial OpenSolaris boot from cd, simplified

# Problems/Challenges (1)

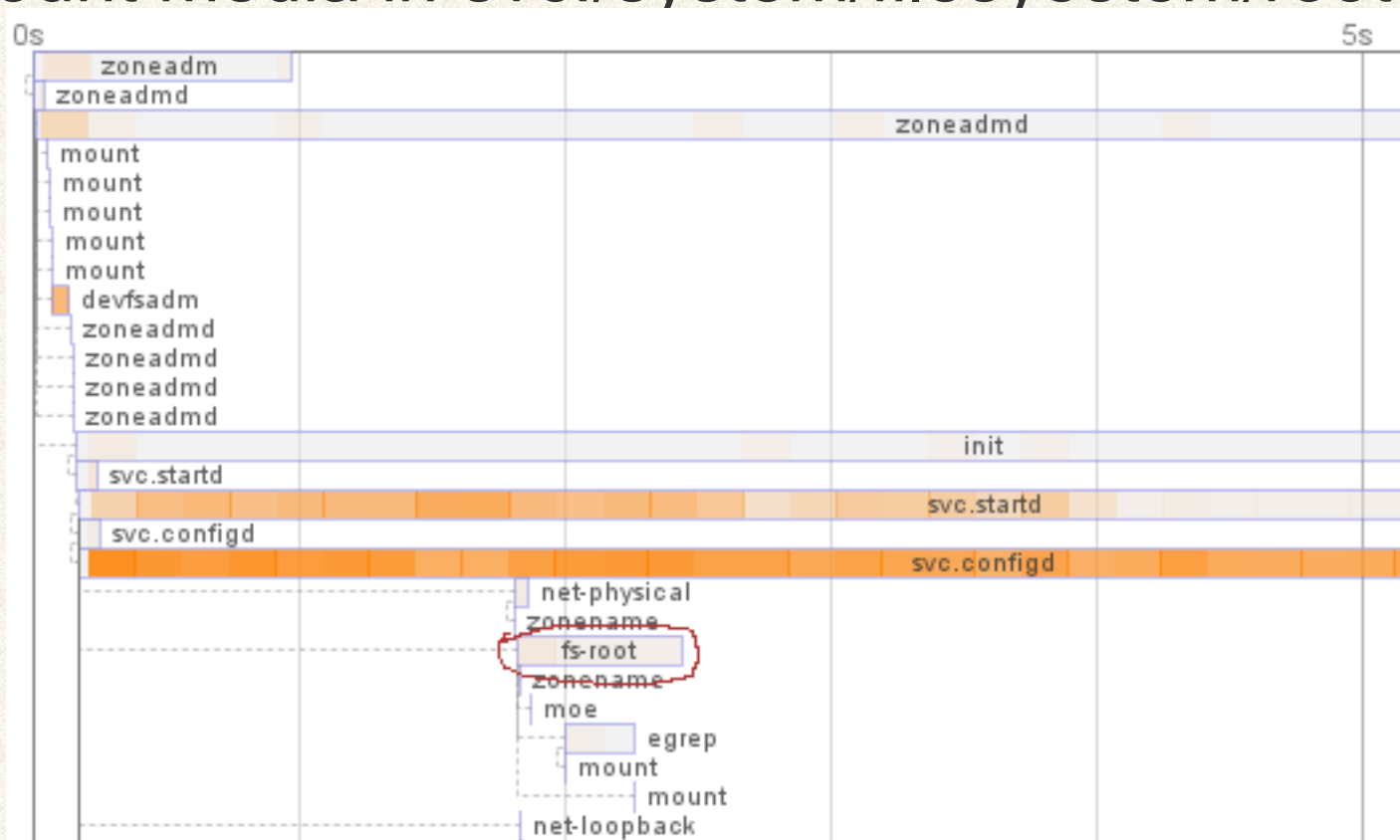
- When to mount the Media (CD, USB ...) ?
- How do we access removable media early in boot ?
- What about SMF's repository.db ?
- Minimizing Ramdisk size (painstaking!)
- Ramdisk files selected by hand.
- Many drivers moved from /kernel/drv to /usr/kernel/drv (eg. audio).
- Device Drivers: Pre-populate /etc/driver\_aliases, /etc/name\_to\_major, /etc/minor\_perm
- Distributions include additional opensource drivers
- Basic /etc/\* configuration files
- Network probing via DHCP

# Problems/Challenges (2)

- How much can we cram in 700MB ?
- Boot from CD/DVD is quite slow.
- Software specific issues
- What about swap ?
- For CD/DVD SMF Parallelism is bad!
- Users expect to be able to save and restore state/configuration in writable USB media.
- Testing!
- Extensive work done to reduce bootup time for CD/DVD.
- Time to boot to Xfce desktop reduced from 10mins to less than 3 mins.

# Analyzing Bootup (1)

- Consider the Bootchart:  
[http://blogs.sun.com/dp/resource/zone\\_boot.png](http://blogs.sun.com/dp/resource/zone_boot.png)
- Mount media in `svc:/system/filesystem/root`



# Analyzing Bootup (2)

- This applies to CD/DVD.
- Data on CD/DVD is arranged on spiral track.
- This is horrible for random access.
- Re-order file data based on their usage frequency.
- DTrace makes life simpler: iosnoop.d from Brendan Gregg's DTraceToolkit provides file access pattern.
- Developed a sliding window algorithm to analyze iosnoop.d data and prepare weighted file list passed to “mkisofs -sort”.
- All the gory details at:  
[http://blogs.sun.com/moinakg/entry/the\\_belenix\\_livecd\\_performance\\_story2](http://blogs.sun.com/moinakg/entry/the_belenix_livecd_performance_story2)

# Analyzing Bootup (3)

- The algorithm attempts to identify durations of heavy random access to multiple files.
- All files in these contention regions are kept close to each other to minimize CD/DVD head movement.
- Contention regions are defined by the number of unique files, arbitrarily fixed at 5 as of today.
- Refer to:  
[http://blogs.sun.com/moinakg/entry/the\\_belenix\\_livecd\\_performance\\_story2](http://blogs.sun.com/moinakg/entry/the_belenix_livecd_performance_story2) for the exact algorithm.

# Analyzing Bootup (4)

- Sample iosnoop.d output showing semi-random access to scattered blocks in different files.

Files being accessed alternately creating a contention region

```
lofil 1877      0 835 R 276344 2048 /usr/foss/lib/libiconv.so.2.2.0 /usr/foss/bin/xfce4-session\0
lofil 3738      0 835 R 276348 2048 /usr/foss/lib/libiconv.so.2.2.0 /usr/foss/bin/xfce4-session\0
lofil 1775      0 835 R 256160 2048 /usr/foss/lib/libxfcegui4.so.3.0.6 /usr/foss/bin/xfce4-session\0
lofil 3534      0 835 R 256164 2048 /usr/foss/lib/libxfcegui4.so.3.0.6 /usr/foss/bin/xfce4-session\0
lofil 1684      0 835 R 256836 2048 /usr/foss/lib/libstartup-notification-1.so.0.0.0 /usr/foss/bin/xfce4-session\0
lofil 3363      0 835 R 256840 2048 /usr/foss/lib/libstartup-notification-1.so.0.0.0 /usr/foss/bin/xfce4-session\0
lofil 1573      0 835 R 107764 2048 /usr/X11/lib/libSM.so.6 /usr/foss/bin/xfce4-session\0
lofil 3138      0 835 R 107768 2048 /usr/X11/lib/libSM.so.6 /usr/foss/bin/xfce4-session\0
lofil 1969      0 835 R 107844 2048 /usr/X11/lib/libICE.so.6 /usr/foss/bin/xfce4-session\0
lofil 3923      0 835 R 107848 2048 /usr/X11/lib/libICE.so.6 /usr/foss/bin/xfce4-session\0
lofil 1533      0 835 R 257456 2048 /usr/foss/lib/libgtk-x11-2.0.so.0.800.6 /usr/foss/bin/xfce4-session\0
lofil 3056      0 835 R 257460 2048 /usr/foss/lib/libgtk-x11-2.0.so.0.800.6 /usr/foss/bin/xfce4-session\0
lofil 1640      0 835 R 263600 2048 /usr/foss/lib/libxfce4util.so.1.0.9 /usr/foss/bin/xfce4-session\0
lofil 3250      0 835 R 263604 2048 /usr/foss/lib/libxfce4util.so.1.0.9 /usr/foss/bin/xfce4-session\0
lofil 1886      0 835 R 264008 2048 /usr/foss/lib/libgdk-x11-2.0.so.0.800.6 /usr/foss/bin/xfce4-session\0
lofil 3719      0 835 R 264012 2048 /usr/foss/lib/libgdk-x11-2.0.so.0.800.6 /usr/foss/bin/xfce4-session\0
```

# BeleniX CDROM Layout

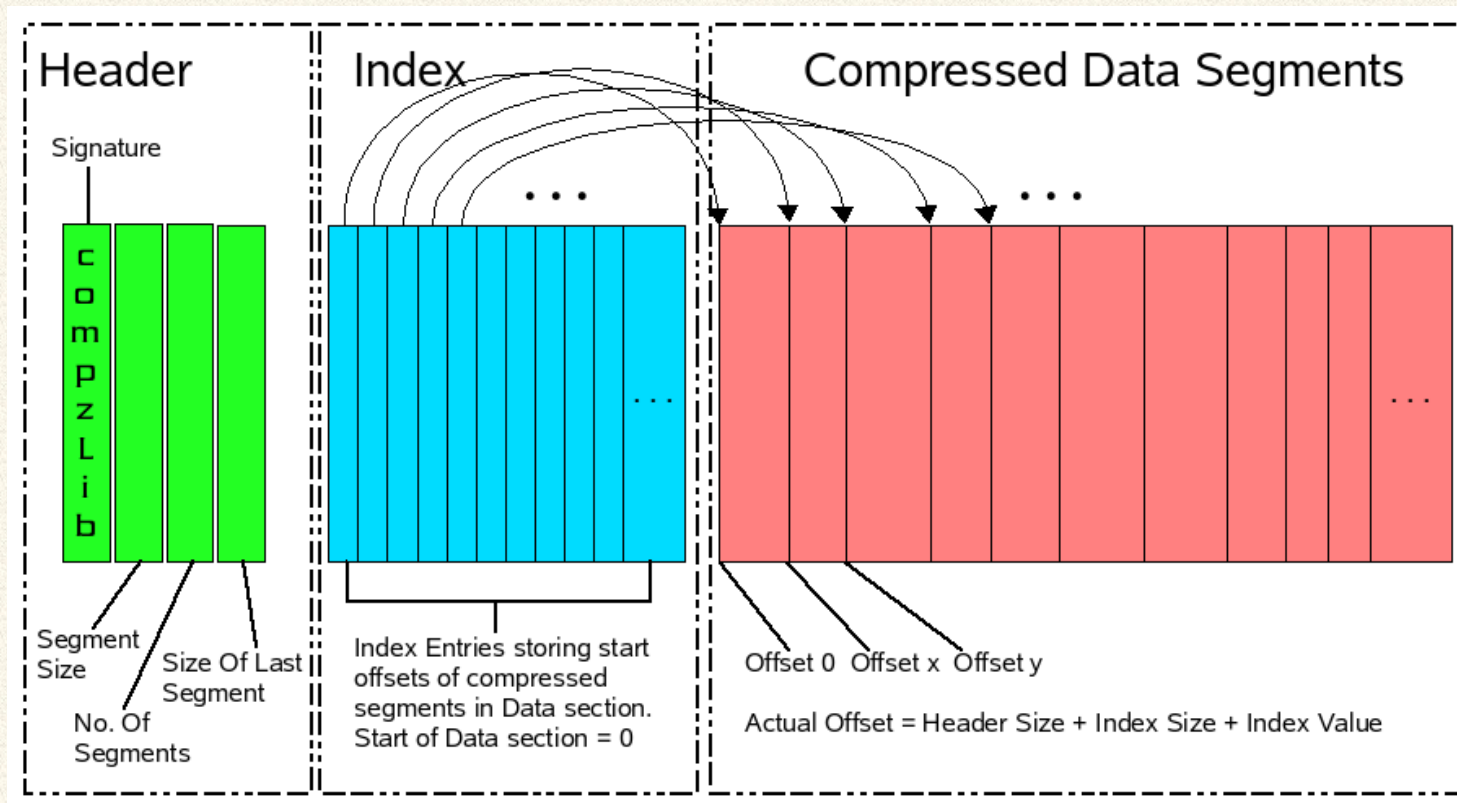


# Cramming Data in a CD

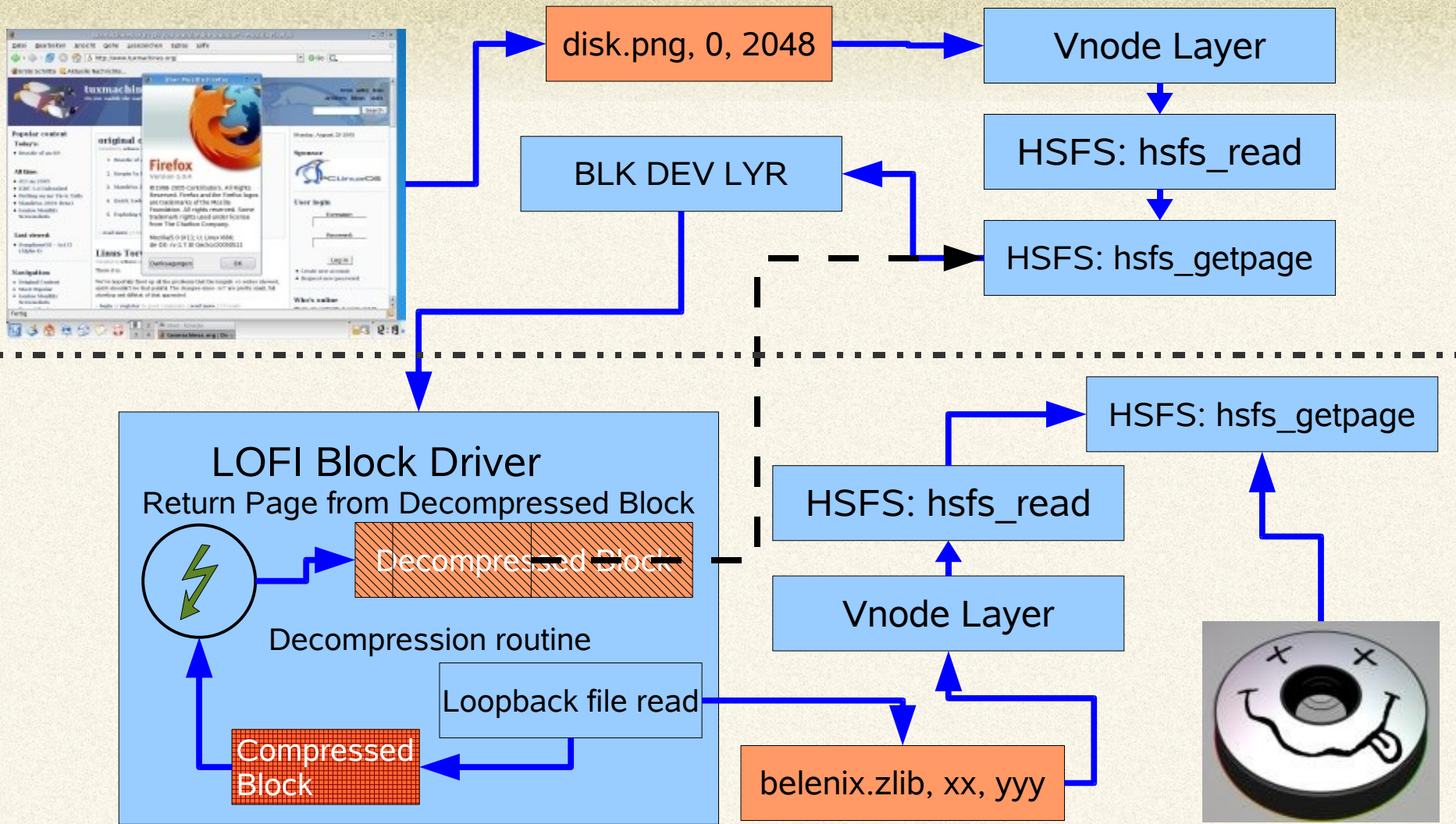
- Introduced zlib compression in the lofi(7D) module.
- /usr and stuff under /opt are put in a hsfs filesystem in a file and the file compressed.
- The compressed file is stored on the CD.
- Lofi provides loopback device for the file.
- Compressed contents are decompressed segment-by-segment on the fly in lofi.
- Zlib decompression code available in-kernel.
- This allows to store 1.8GB of data in 700MB.
- More info: <http://www.belenix.org/?q=compression>

# Lofi Compression

- 128K size file segments are compressed and indexed (see slide notes).



# Transparent Decompression



# HSFS Improvements (1)

- CD/DVD Access time is high: seek time + rotational latency.
- Seek time is the major component
- Random access aggravates seek time issue
- Solution: I/O Scheduling and Readahead
- I/O Scheduler attempts to optimize seeking
- Serialize and re-order I/O requests in a pipeline
- Implementation in BeleniX uses CLOOK algorithm and deadline scheduling.
- Coalesce multiple adjacent I/Os into one I/O
- Readahead benefits sequential access

# HSFS Improvements (2)

- HSFS Readahead does optimistic reads
- Last block of cache-read triggers further background reads
- In ideal sequential access case application never encounters disk I/O lag.
- 4 consecutive adjacent reads used to detect sequential access pattern.

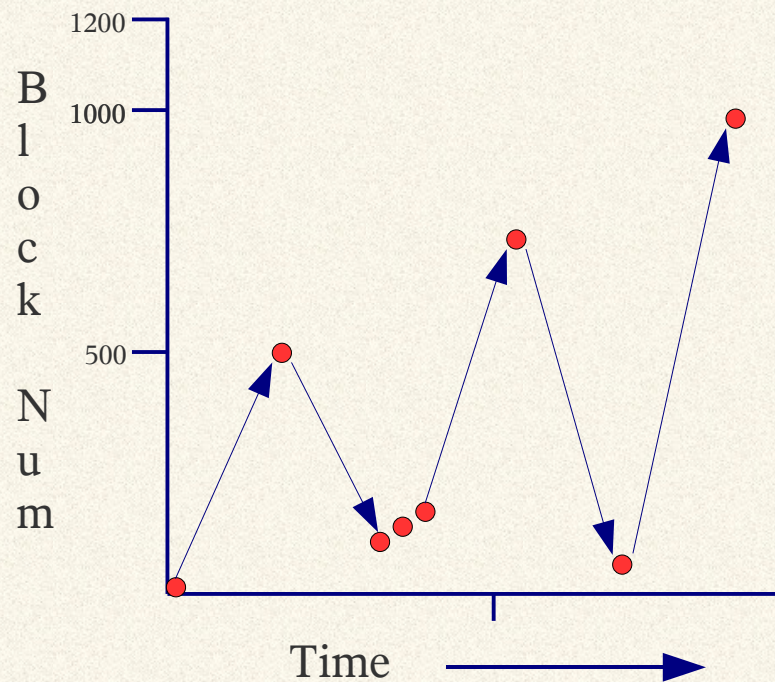
# I/O Scheduling Benefit

Example: Requested disk blocks – 10, 500, 100, 110, 120, 720, 50, 1000

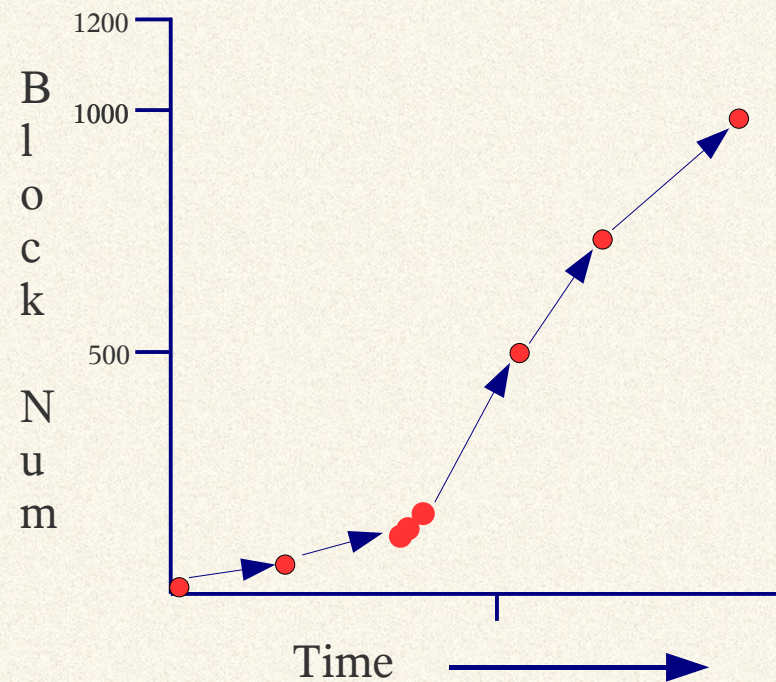
Disk Block Size = 10 bytes

Reordered, Coalesced Disk Blocks – 10, 50, 100-110-120, 500, 720, 1000

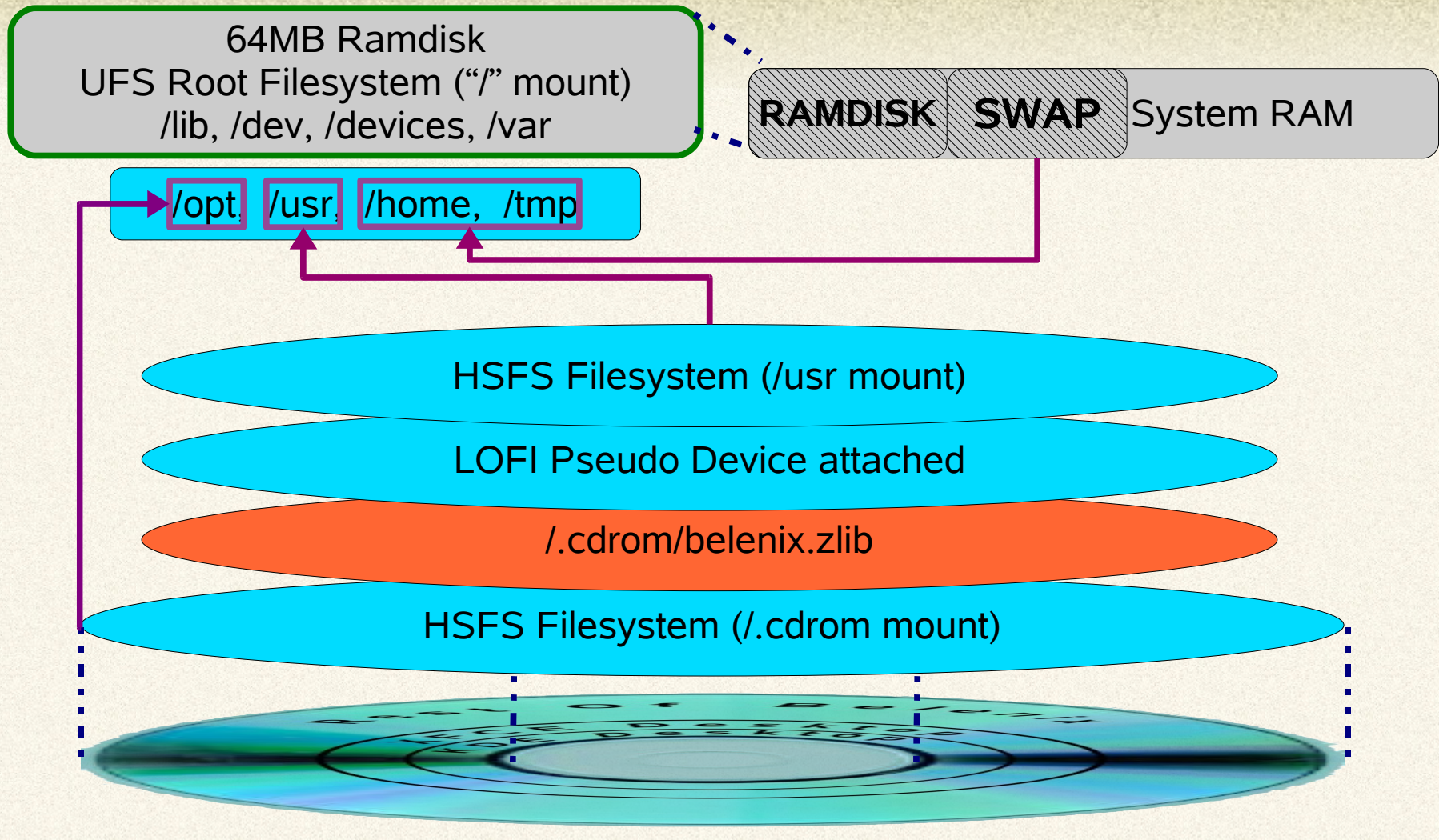
Disk head seek without I/O Scheduling



Disk head seek with I/O Scheduling



# Filesystem Organization



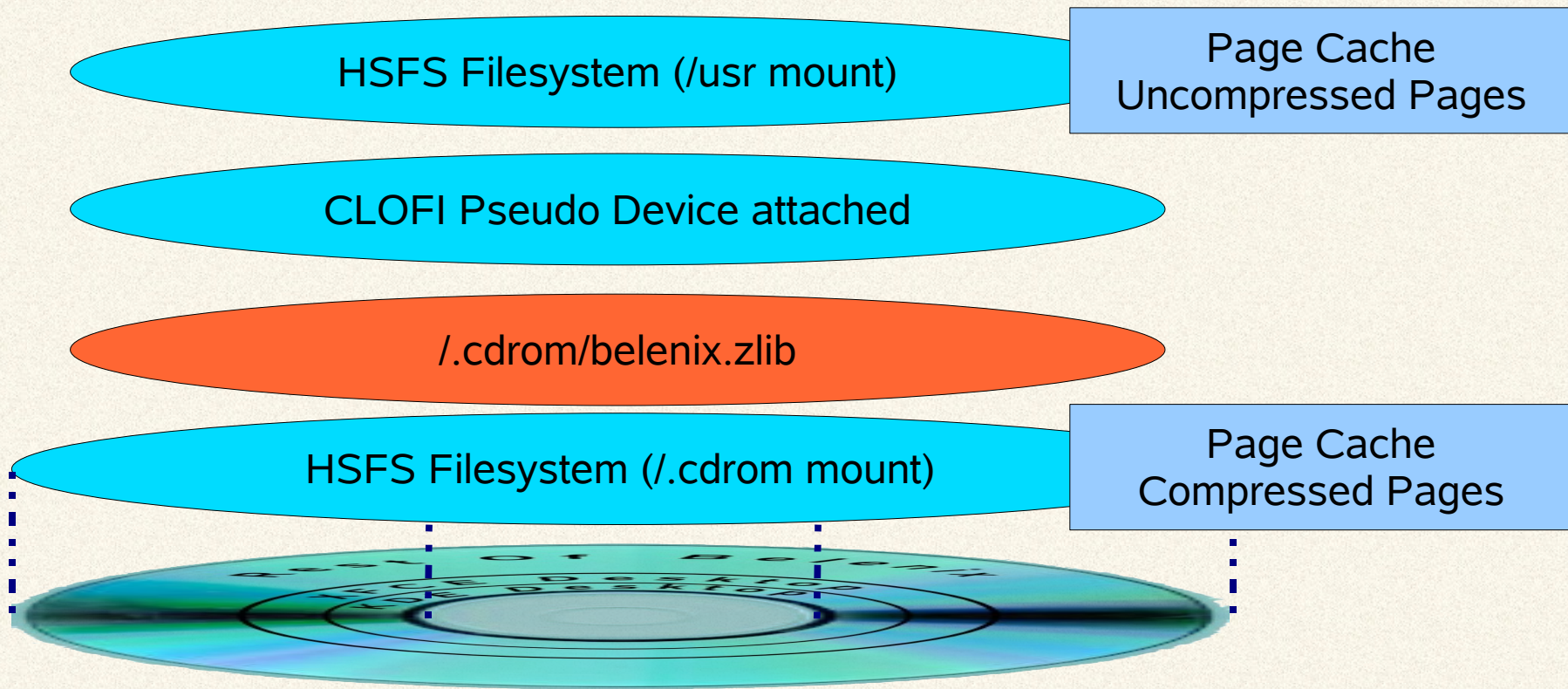
# Other changes (1)

- CD/DVD is mounted in fs-root.
- CD/DVD is detected via libdevinfo as /dev links are not present at this time.
- fstyp(1M) is used to check the volume label to determine which media to use.
- Desktop choice and keyboard selection are also done in fs-root
- Additional mountable harddisk partitions are detected in svc:/system/device/local service.
- An interactive curses-based Xorg configuration UI is present in BeleniX to tweak Xorg settings.

## Other changes (2)

- Disable a bunch of additional services. The generic\_live.xml smf profile in OpenSolaris LiveKit (<http://www.opensolaris.org/os/project/livemedia/>)
- Simplified startkde and startxfce scripts. Eg. Avoid calling “which ssh-agent” - reduces PATH lookup.
- For KDE use a prebuilt ksycoca config cache.
- Prebuilt font caches via fc-cache.
- Use full pathnames in startup scripts to reduce PATH lookup.
- Use a modified pagein utility to preload libs.
- Modified pagein touches one page in every 128K file segment.

# Double Caching



# Bootable USB media

- Ability to boot from USB Flash media (PenDrive) or USB harddisk – similar to CD/DVD boot.
- BIOS Support required to boot from USB storage.
- USB storage media is detected via a similar mechanism for CD/DVD media.
- libdevinfo(3LIB) is used to scan for USB storage having a UFS filesystem and mounted.
- A script is present to format a USB storage media and dump BeleniX onto it – works for Solaris Express LiveDVD as well.
- Changes for USB boot were done by a 3<sup>rd</sup> year degree student.

# USBDump script operation (1)

- 1) Disable `volfs` or `rmvolmgr` as this can interfere with raw removable device access.
- 2) Mount the BeleniX ISO image.
- 3) Use `rmformat(1)` to get a list of removable devices.
- 4) Display this list for User – Selection
- 5) Determine capacity of selected device via `fdisk -W`
- 6) Compute sizes of backup and root slices for creating UFS filesystems.
- 7) Create a default full-disk partition using `fdisk -B`
- 8) Create slices using `rmformat -s`
- 9) Format the root slice using `newfs(1M)` and mount it
- 10) Copy entire contents of BeleniX ISO to the root slice.

# USBDump script operation (2)

- 11) Patch boot/grub/menu.lst on USB to remove harddisk boot option (see notes).
- 12) Modify the miniroot copied onto the USB device to include a flag file indicating LiveUSB behavior.
- 13) Run `installgrub(1M)` to install GRUB onto USB media.
- 14) Unmount the various mounts.
- 15) Re-enable `volfs` or `rmvolmgr`

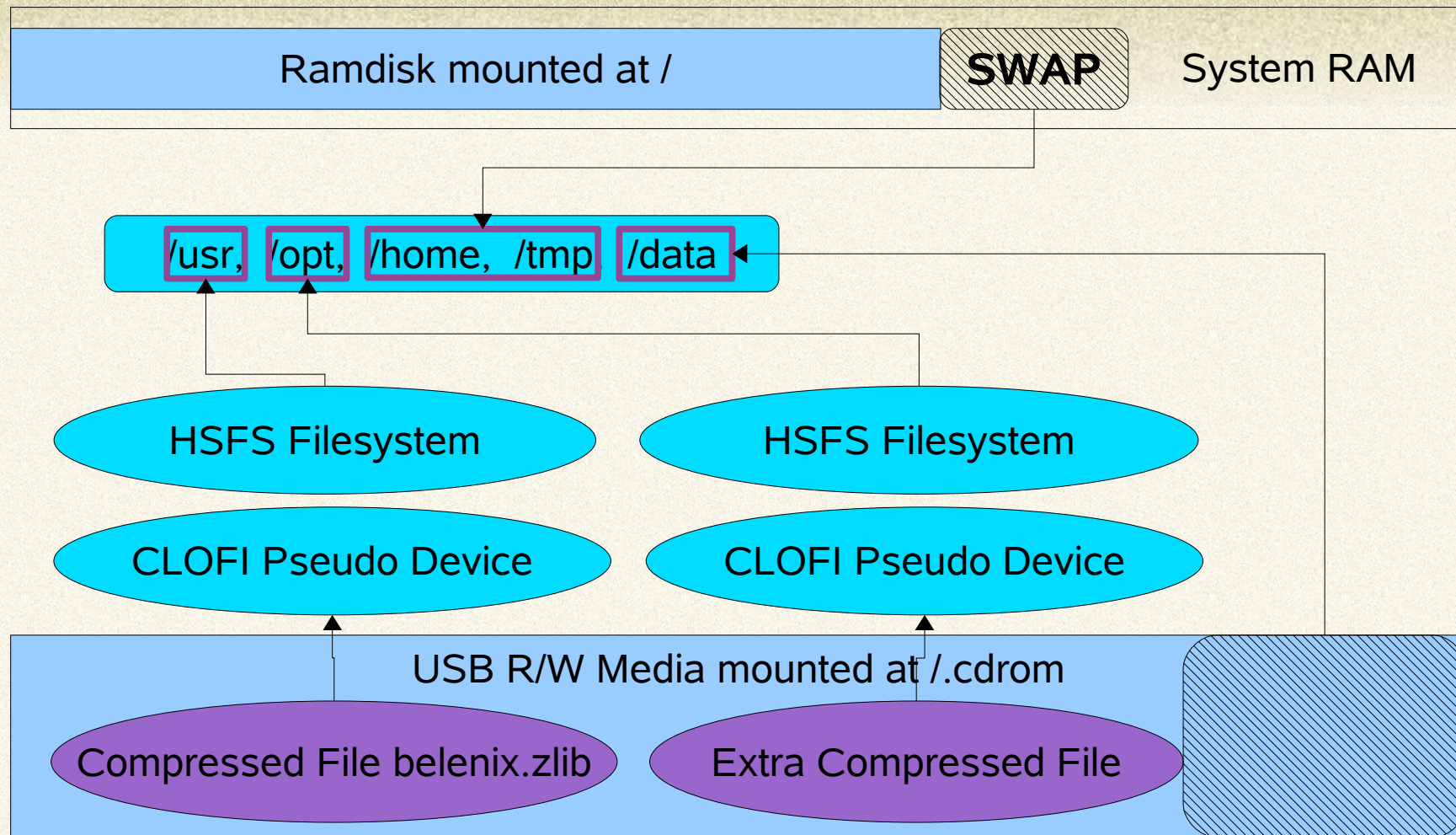
# USB media advantages

- Writable USB media can store configuration and session state.
- State can be preserved during shutdown and/or at periodic intervals.
- Session and user data can be encrypted (using xlofi or ZFS encryption).
- Carry your entire Operating Environment (OS, Apps, Data) in your pocket.
- Ability of user to add applications.
- Add VPN Client/remote office/remote desktop connectivity software.

# USB media limitations

- Requires a computer to be available
- The BIOS should support booting off USB storage
- Hardware compatibility issues in the OS can cause problems (also with LiveCD/DVD).
- The number of supported writes to internal pages of a USB flash media is limited (typically 100000).
- What if I lose my little USB Pen Drive!

# USB media layout



# Remastering Live Media

- Live Media support can allow one to create own customized distro.
- It can be fun, it can be useful.
- Software demo CDs, System Recovery CDs etc.
- BeleniX allows remastering via the Remastering Kit.
- For OpenSolaris the LiveKit is available to create bootable DVD images from a Solaris Express DVD ISO image.
- A LiveMedia OpenSolaris Project has been created to get all the livemedia technologies into OpenSolaris.

# OpenSolaris LiveKit

- This is part of the OpenSolaris Live Media project: <http://www.opensolaris.org/os/project/livemedia/>
- This generates a Live DVD image given the ISO image of a supported build.
- The scripts contain detailed comments.
- It does an alternate-root install of the SUNWCxall package cluster.
- It then generates a LiveDVD image from the alternate root.

# Current LiveCD/DVD usage

- The Solaris Express LiveDVD opens up possibilities for other product groups in SUN.
- For eg. the Java Enterprise System (JES) team considering a JES LiveDVD.
- Potential to reduce JES demo setup time from days to hours!
- Live DVD is the long term install strategy for SUN Solaris.
- A bootable DVD provides a better user experience during installation.
- Users get a feel of the OS before installing.

# Next Steps (1)

- Integrate the code and script changes back into OpenSolaris.
- The startup script changes will require further discussion/thought.
- The ramdisk is small (64MB) and files selected/maintained manually.
- Need to automate the tedious work of generating the ramdisk file list – minimal package selection is not enough.
- Make the lofi compression changes largefile-aware.

## Next Steps (2)

- lofi compression is designed as endian-neutral and will require 64bit byteswap macros.
- The basic feature of UnionFS is very useful for Live Media.
- Allows pseudo writes to read-only files – upgrade existing software.
- Work going on to implement session/state persistence on writable USB media.
- Support for encrypting persistent data is also planned via xlofi.

## Next Steps (3)

- OpenSolaris Recovery Toolkit: A customised LiveCD packed with OpenSolaris recovery tools and scripts.
- Ability to use Linux swap without corrupting it.
- Linux swap keeps a one-page header in the swap area; remaining area is unformatted and can be used by OpenSolaris.
- On machines with lots of RAM ( $\geq 2\text{G}$ ) the entire LiveCD contents can be loaded into a big ramdisk.
- Execution off a ramdisk is blindingly fast.

## Next Steps (4)

- BeleniX has a simple harddisk install mechanism that extracts the CD contents to harddisk.
- Same can be applied to Solaris Express to get an easy and fast install path.
- A GUI for remastering/customizing your own LiveCD.
- A Busybox-like shell for reducing bootup scripts overhead.
- Will contain basic built-in versions of many Unix commands – good for harddisk boot as well.
- ...

# Some Resources

- OpenSolaris LiveMedia Project:  
<http://www.opensolaris.org/os/project/livemedial/>
- BeleniX Docs:  
[http://www.genunix.org/distributions/belenix\\_site/?q=docs](http://www.genunix.org/distributions/belenix_site/?q=docs)
- DTrace Toolkit:  
[http://www.brendangregg.com/dtrace.html#DTrace Toolkit](http://www.brendangregg.com/dtrace.html#DTraceToolkit)
- OpenSolaris New Boot Architecture:  
[http://blogs.sun.com/setje/entry/a\\_new\\_boot\\_architecture](http://blogs.sun.com/setje/entry/a_new_boot_architecture)



# LiveMedia Technologies for OpenSolaris

Thank You