

# open



USE



IMPROVE



EVANGELIZ  
E

## Deep Dive into Project Indiana

Moinak Ghosh  
SUN Microsystems India

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



# What is OpenSolaris ?



# opensolaris

Search

## About OpenSolaris

- Project Overview
- FAQ Center
- Roadmap
- Governing Board
- Site Map

## Communities

- Portal
- Nevada Community
- All Communities

## Projects

- Portal
- All Projects

## Code

- Source Browser
- Download
- Tools
- Bug Database

## Connect

- Register
- Mail Lists
- Jive Forums
- Documentation
- Blogs
- Planet
- Announcements
- Events
- News
- Related Links



## What is the OpenSolaris Project?

The OpenSolaris project is an open source community and a place for collaboration and conversation around OpenSolaris technology.

The community represents a wide variety of people around the world, including developers adding functionality to the system or customizing the technology for new applications and platforms; system administrators implementing Solaris technology in data centers; educators and students researching operating systems in universities; and new users exploring the technology and discovering that OpenSolaris offers new opportunities.

The OpenSolaris source code is already cutting edge, but innovation happens everywhere so we welcome your involvement. Here's what our community has to offer:



Start by taking a look at our [project overview](#) and [Roadmap](#). If you need help in getting OpenSolaris installed, see the resources on our [Installing OpenSolaris](#) page. (If you're looking for information about the Solaris Operating System, [go over here](#)).



Join dozens of lively [discussion forums](#) for conversations on almost any topic. If you need help installing OpenSolaris, you can join our [opensolaris-help](#) forum. To keep abreast of what's happening in the community you may wish to subscribe to [opensolaris-announce](#), or join the fray in [opensolaris-discuss](#). You can also join the conversation via [IRC](#) on [irc.freenode.net](#).



Visit our [Community Portal](#) to learn all about how to get involved. Or check in on [Nevada](#).



Access our [Projects Portal](#) to explore active projects we're hosting.



[Download](#) the source, BFU archives and tools you'll need. HTTP and Torrent downloads are offered.



[Explore the source](#) using our 'wicked fast' code browser, [OpenGrok](#).

### OpenSolaris Starter Kit

The free (no shipping or handling fees) Starter Kit includes three OpenSolaris distributions and step-by-step instructions on how to install and boot.  
» [Request a DVD](#)  
» [Order Status](#)

### User Groups

If you are new to OpenSolaris and are interested in participating in a user group, visit the [User Group community](#).

### Squawk

Chat about OpenSolaris using IRC at [irc.freenode.net](#). Now featuring [nine](#) chat rooms.







# What are the community issues ?

- opensolaris.org grew up around the source, and engaging developers
  - Not necessarily a good showcase for OpenSolaris technology
  - Poor download experience
  - Little user community growth



# What are the technology issues ?

- High adoption barriers
  - Install, Packaging, Hardware
- Media size has grown too large to reach important markets
- No clear binary technology base to work from
- No formal roadmap or public process for SXCE/SXDE



# Enter Project Indiana (1)

- OpenSolaris reference binary distribution
- SUN and Community jointly built
- Single CD install and network package repository
- 100% re-distributable
- LiveCD/DVD functionality
- ZFS as default file-system
- More intuitive update experience with ZFS rollback functionality



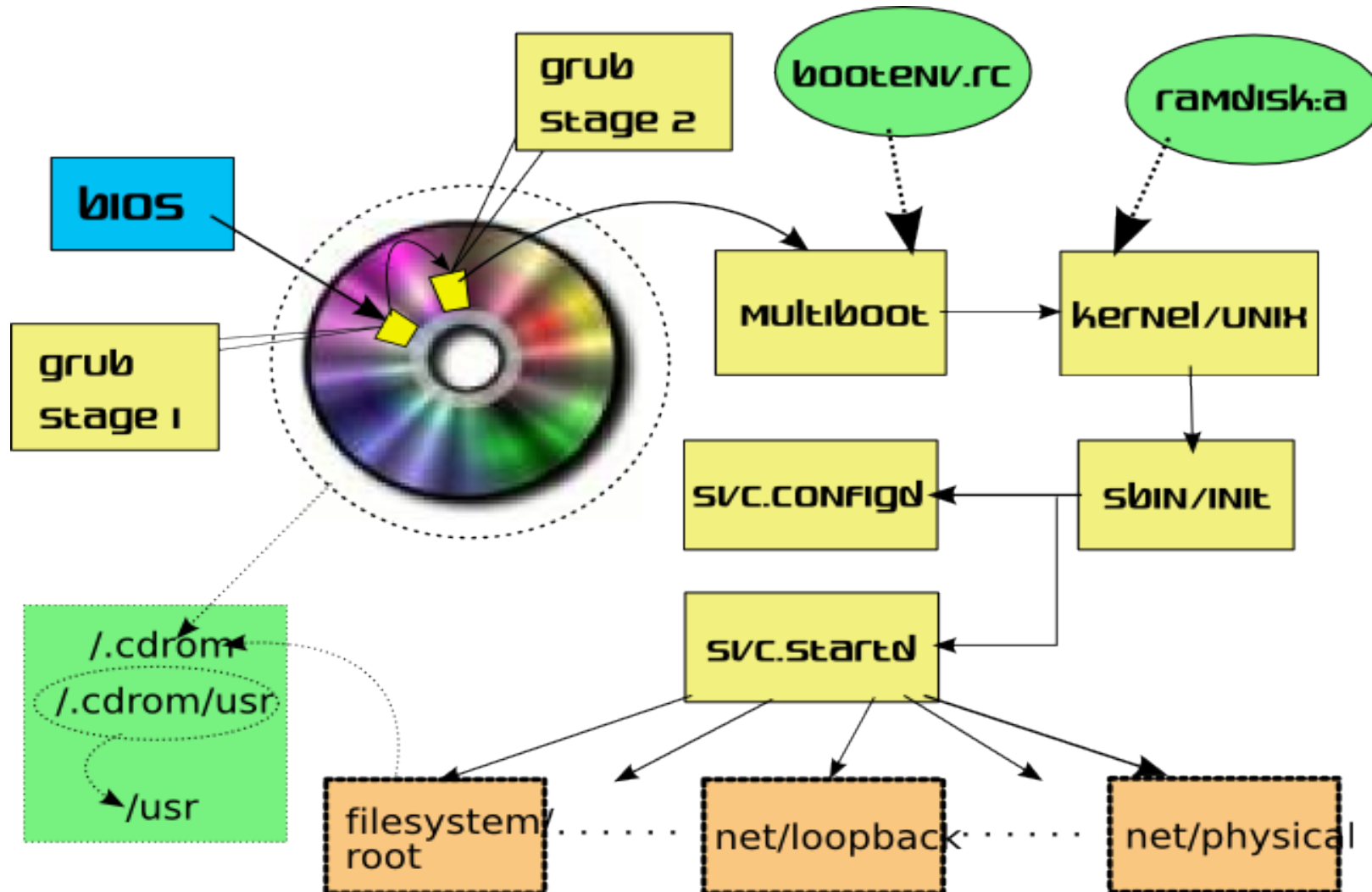
# Major Indiana Technologies

- LiveCD/DVD/USB Infrastructure derived mostly from BeleniX – Slim Install
  - OpenSolaris uses GRUB and is multiboot aware
  - OpenSolaris also uses an Initial Ramdisk
- New Packaging
- Distribution Constructor
- Modern Simple Installer
- Live Upgrade Mechanism

# The Basic Live-boot 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



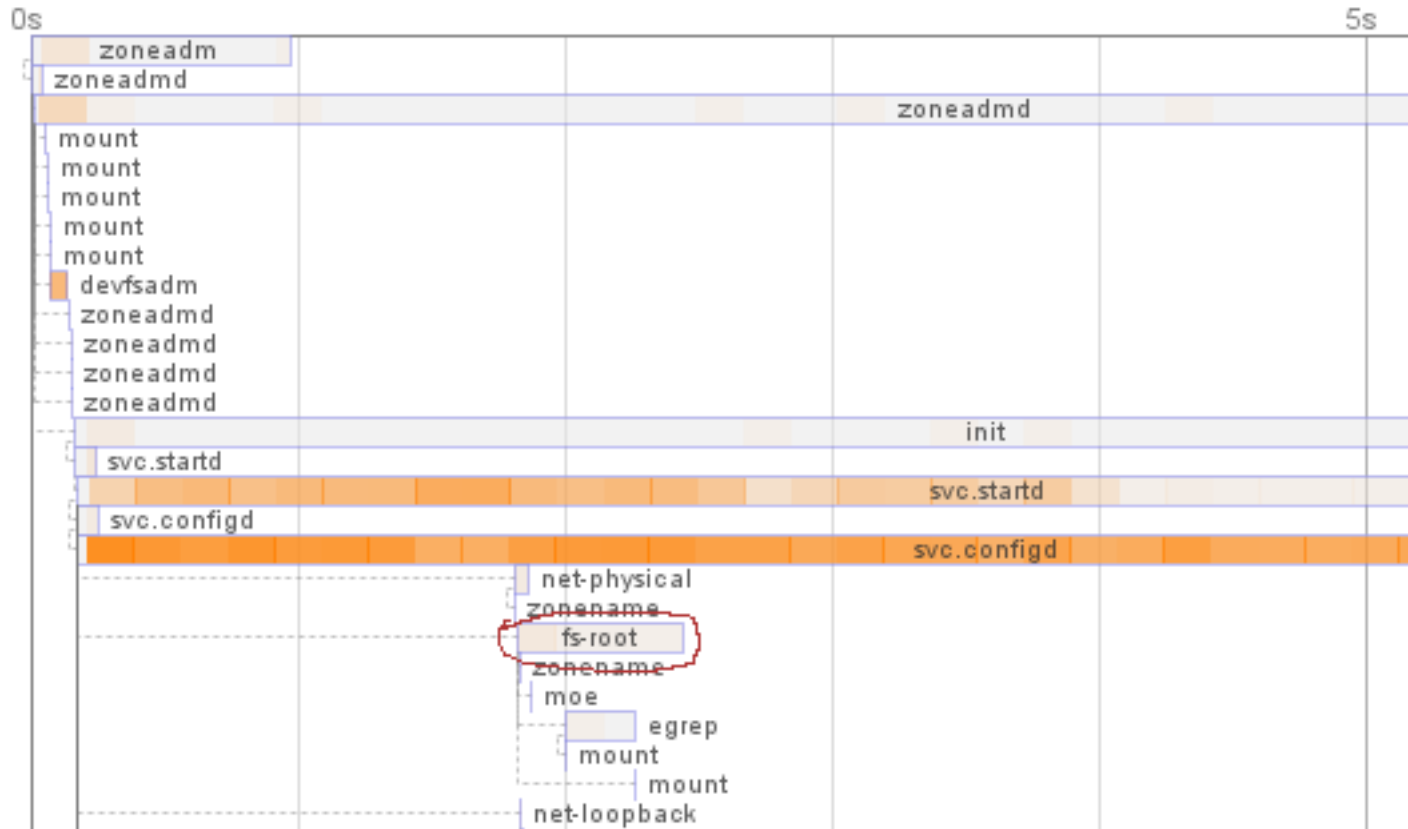
Initial OpenSolaris boot from cd, simplified

# Problems/Challenges

- When to mount the Media (CD, USB ...) ?
- How do we access removable media early in boot ?
- Minimizing Ramdisk size (painstaking!)
- Device Drivers: Pre-populate /etc/driver\_aliases, /etc/name\_to\_major, /etc/minor\_perm
- Basic /etc/\* configuration files
- Network auto-probing via DHCP
- How much can we cram in 700MB ?
- Extensive work done to reduce bootup time for CD/DVD.
- What about swap ?

# 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)

- 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”.
- The algorithm attempts to identify durations of heavy random access to multiple files
- 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)

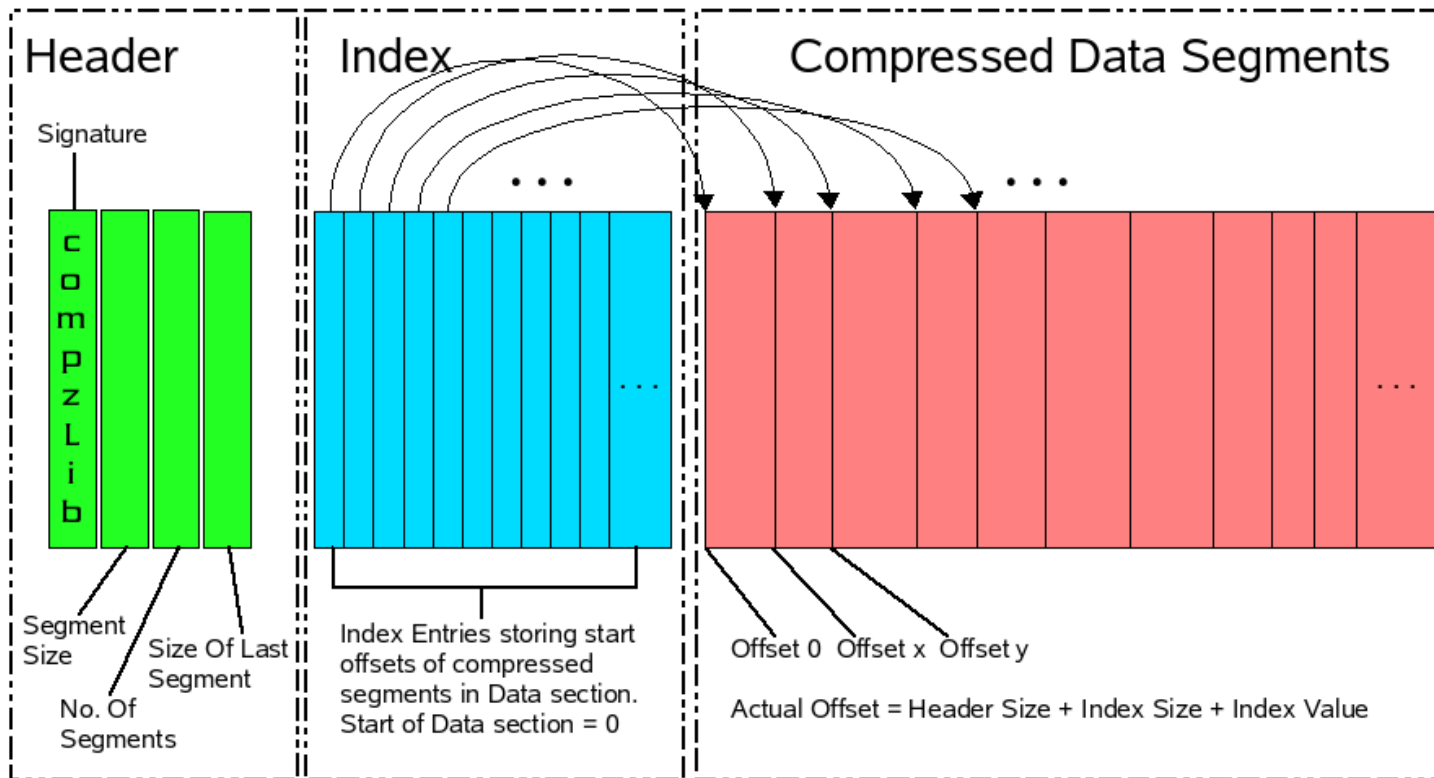


# Cramming Data in a CD

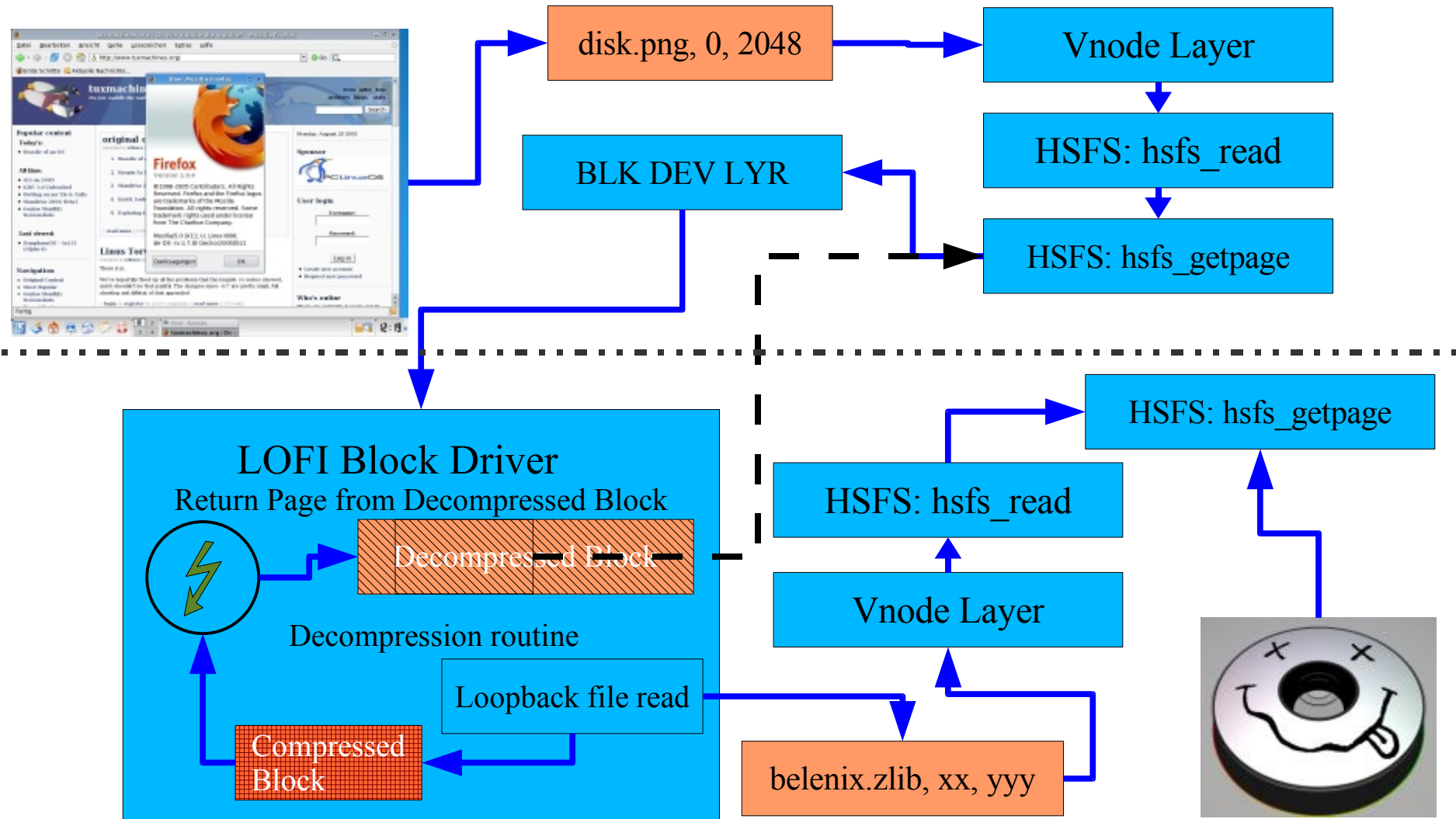
- Introduced zlib compression in the lofi(7D) module – essentially loopback compression
- /usr and stuff under /opt are put in a hsfs filesystem in a file and the file compressed.
- Compressed contents are decompressed segment-by-segment on the fly in lofi loopback driver.
- Zlib decompression code available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

# Lofi loopback Compression layout

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



# Transparent loopback Decompression





# I/O Scheduling for CD/DVD

- 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 originally in BeleniX uses CLOOK algorithm and deadline scheduling.
- Coalesce multiple adjacent I/Os into one I/O
- Readahead benefits sequential access

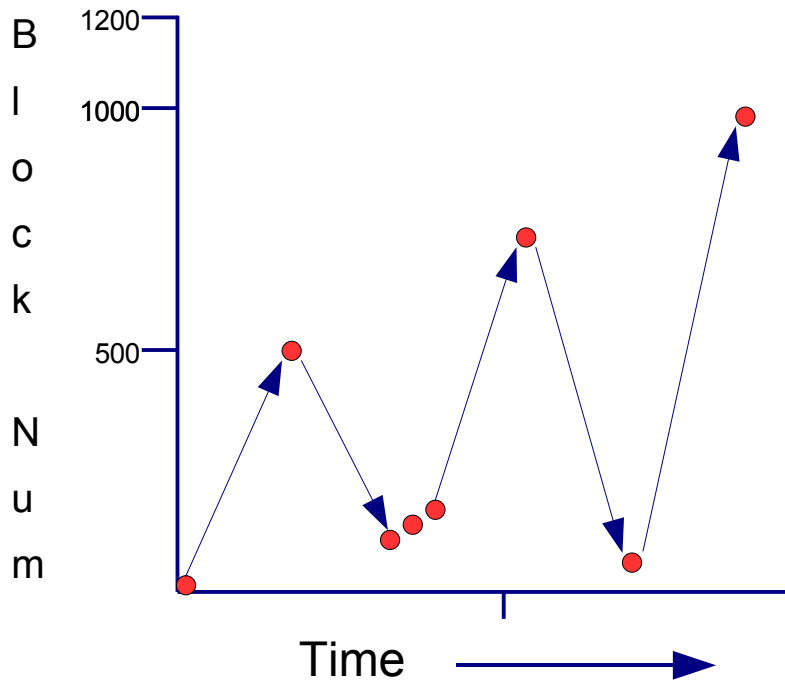
# I/O Scheduling Benefit

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

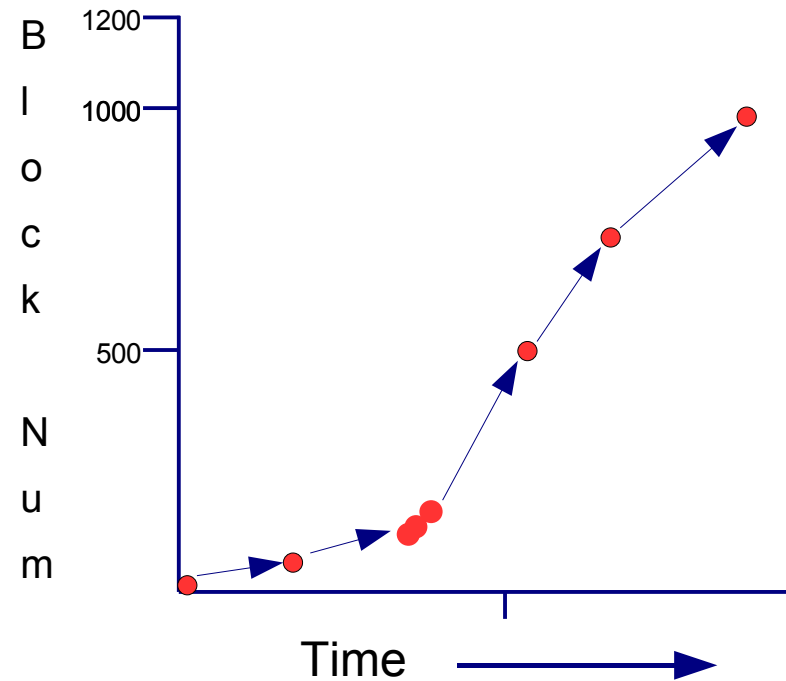
Disk Block Size = 10 bytes

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

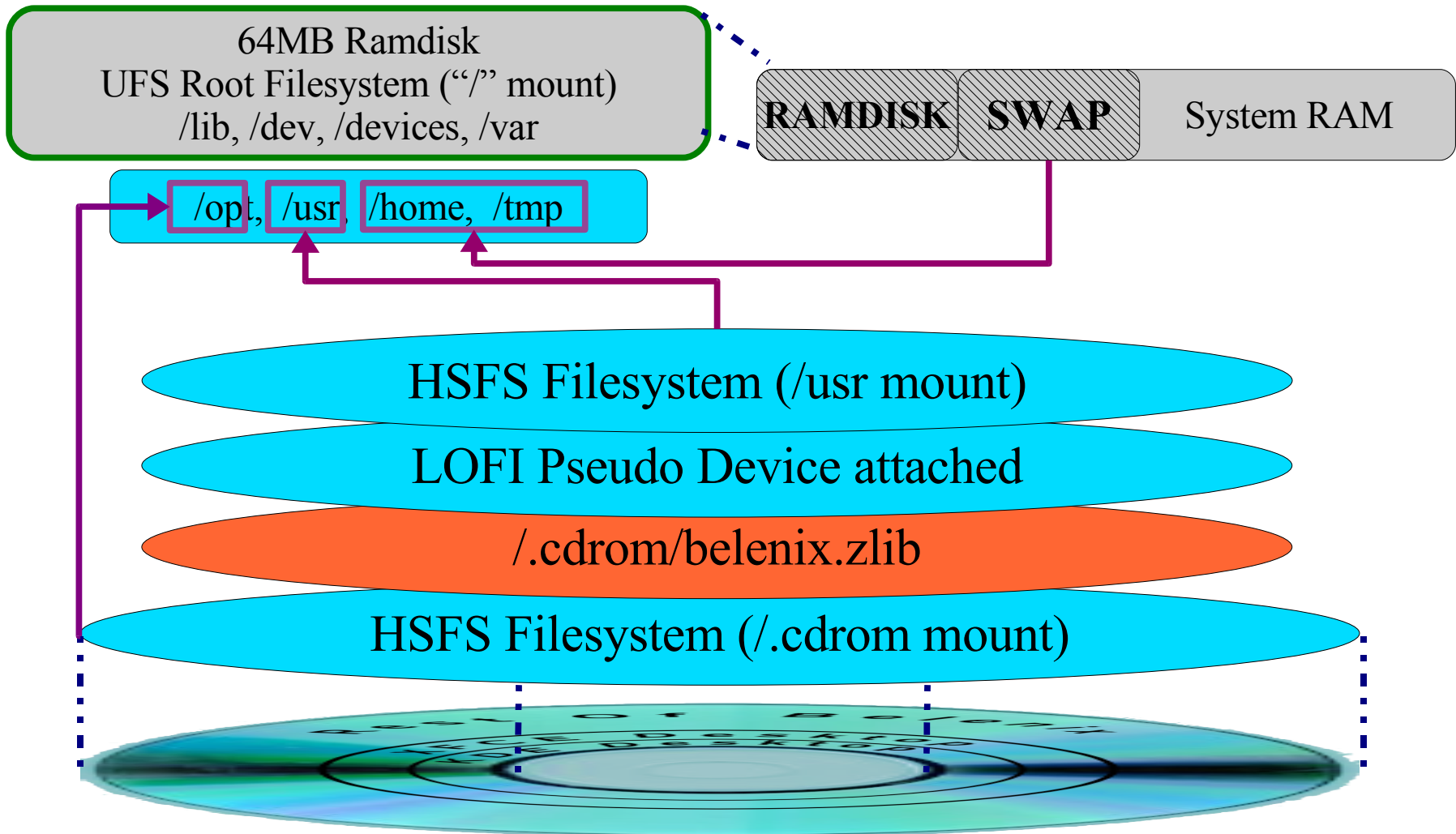
Disk head seek without I/O Scheduling



Disk head seek with I/O Scheduling



# Filesystem Organization





# OpenSolaris boot from USB

- 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 LiveCD onto it – first used in BeleniX
- Changes for USB boot were done by Anil Gulecha while he was learning OpenSolaris from scratch.



# Indiana Demo

- Booting and Install off CDROM – new installer
  - Note: Actual install runs parallel to rest of the preso.
- Booting off USB
- How USB images are created



# How do we construct the ISO (1)

- LiveCD is built out of root image of an installed OS instance.
- Alternate-root install capability of OpenSolaris is used to create an OS installation in a directory.
- Patch base root image with addition configuration and scripts for LiveCD operation.
- Perform post-install configuration on the alternate-root, eg: run fc-cache
- Create a UFS filesystem in a file – this will be our minimal ramdisk. The files in the ramdisk are listed in `mkbootcd.files.minimal`.
- Create essential symlinks on `/dev` in the ramdisk image.



## How do we construct the ISO (2)

- Use `gzip -9` to compress the ramdisk image. This is decompressed by GRUB at load time.
- Create CDROM filesystem images using `mkisofs` for `/usr` and `/opt` directories.
- These images are compressed using `lofi` compression.
- Build the final LiveCD ISO off the alternate root.
- Tools used:
  - `Mkisofs` from `cdrtools`
  - `Gzip`, `UFS` tools
  - `Lofi` compression utility

# The Live kit and Distro Constructor

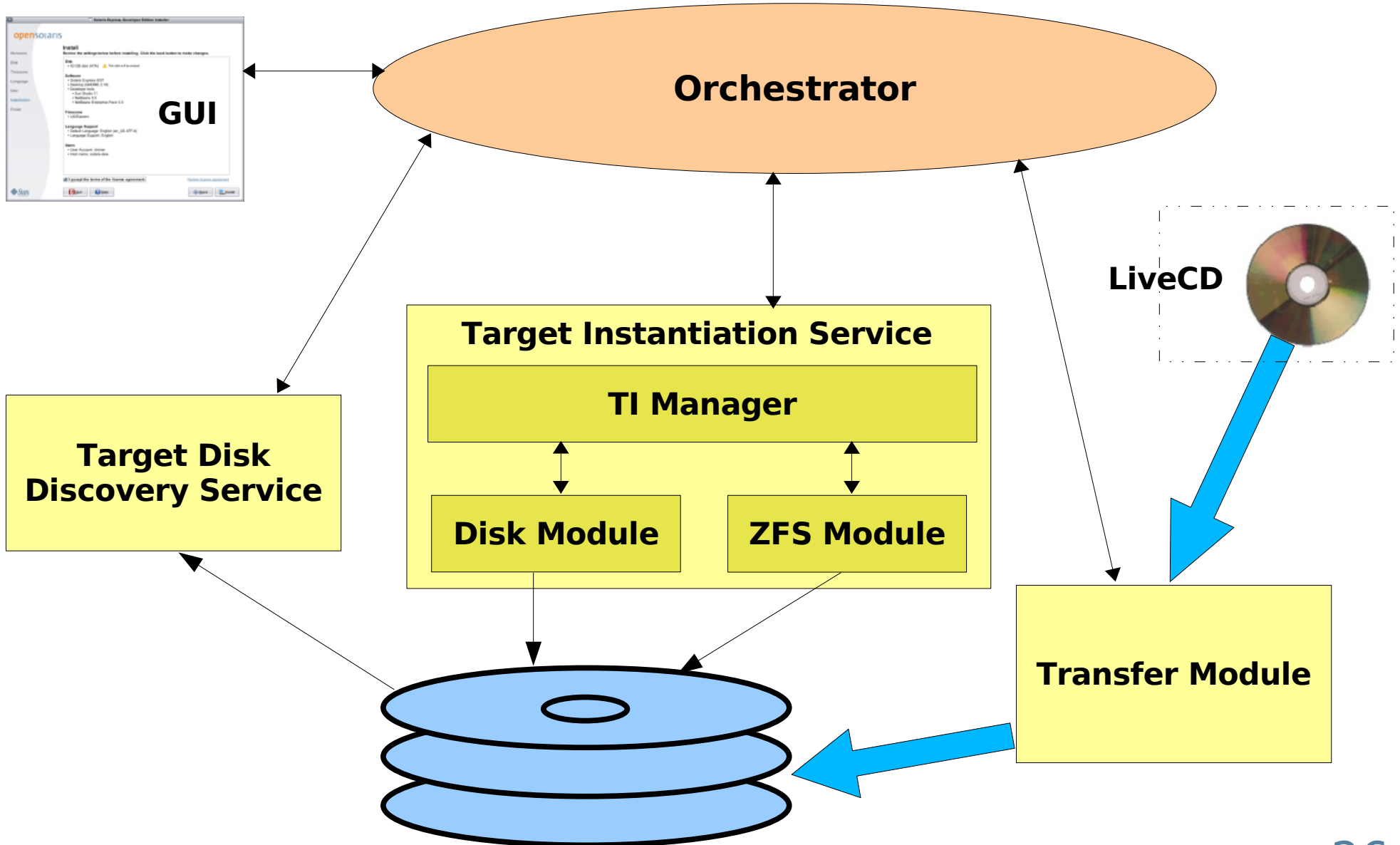
- The Live kit was developed out of the BeleniX LiveCD scripts.
- Browse the source
  - <http://cvs.opensolaris.org/source/xref/livemedia/livemedia/>
- Get Like source:
  - `hg clone ssh://anon@hg.opensolaris.org/hg/livemedia/livemedia`
- Participate: `livemedia-discuss-subscribe@opensolaris.org`
- The Distro Constructor is derived out of the Live Kit but uses the new packaging.
  - `hg clone ssh://hg.opensolaris.org/hg/caiman/distro_constructor`
- A more organized approach to distro creation
- Make it easy to create your own distro



# Install from LiveCD – Slim Install

- Setup ZFS root on harddisk
- Get timezone, root passwd and one userid/passwd
- Transfer Live Image to harddisk
- No package operations, cpio is used - fast!
- Perform postinstall configuration
- Major components
  - Caiman Installer Frontend
  - Target Instantiation
  - Transfer Module
- See: [http://www.opensolaris.org/os/project/caiman/Slim\\_Install](http://www.opensolaris.org/os/project/caiman/Slim_Install)
- Source: hg clone `ssh://anon@hg.opensolaris.org/hg/caiman/slim_source/`
- Browse Source:  
[http://src.opensolaris.org/source/xref/caiman/slim\\_source](http://src.opensolaris.org/source/xref/caiman/slim_source)

# Slim Install Architecture



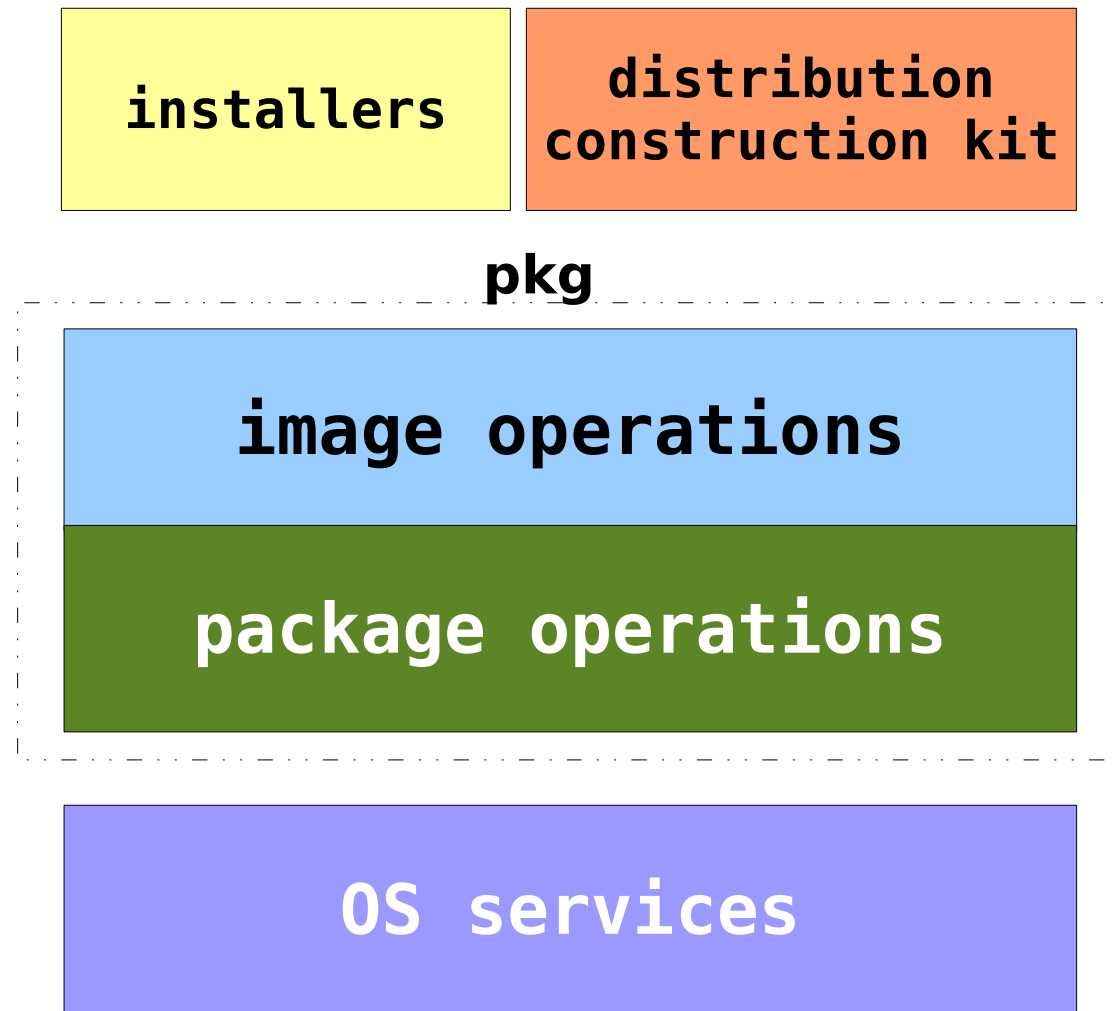


# New Packaging

- No scripting in packages – exhaustive metadata
- No build system in packaging – packaging only lays out binary bits on the system
- Dynamic package refactoring – sub-package file tagging and filtering capability
- Support concept of filesystem images
- Images can be linked and updated in parallel – supports virtualization needs
- Per-user images
- ZFS aware – snapshots and rollback
- Get the source:
  - `hg clone ssh://anon@hg.opensolaris.org/hg/pkg/gate`

# New Packaging - Layering

- Package system will provide image mgmt APIs, as well as more primitive package ops
- Distro kit being designed alongside new installer





# How do I get involved ? (1)

- Subscribe to mailing lists
- Friendly community
- Browse project pages, suggest and discuss new features
- Browse source, check-out Mercurial repositories
- Create your own distro, provide feedback
- Love Python ? Hack around with Pkg.



## How do I get involved ? (2)

- Indiana Core
  - [indiana-discuss@opensolaris.org](mailto:indiana-discuss@opensolaris.org)
  - <http://www.opensolaris.org/os/project/indiana/>
- Caiman Installer, Slim Install project
  - [caiman-discuss@opensolaris.org](mailto:caiman-discuss@opensolaris.org)
  - <http://www.opensolaris.org/os/project/caiman/>
  - [http://www.opensolaris.org/os/project/caiman/Slim\\_Install/](http://www.opensolaris.org/os/project/caiman/Slim_Install/)
- New Packaging
  - [pkg-discuss@opensolaris.org](mailto:pkg-discuss@opensolaris.org)
  - <http://www.opensolaris.org/os/project/pkg/>
- Distro Constructor
  - <http://www.opensolaris.org/os/project/caiman/Constructor/>



# Roadmap, TODOs, Opportunities (1)

- New project: Snap Upgrade – Safely upgrade a live system via ZFS snapshots and then reboot to the new snapshot.
  - [http://www.opensolaris.org/os/project/caiman/Snap\\_Upgrade/](http://www.opensolaris.org/os/project/caiman/Snap_Upgrade/)
- Lots of stuff to be done in Distro Constructor
  - Support package and automatic ramdisk customization
  - GUI utility like Fedora Revisor, GUI for branding modifications
  - Reduce ramdisk size, improve bootup time further
  - Generate net-install images
  - Build LiveCD variants: Games CD, Rescue CD etc.
  - Auto-check package deps
  - Support for creating distro profiles



## Roadmap, TODOs, Opportunities (2)

- Enhance installer. Roadmap here:
  - <http://www.opensolaris.org/os/project/caiman/Roadmap/>
- Port FOSS stuff to OpenSolaris using Pkgbuild.
- It builds OpenSolaris packages from RPM spec files:
  - <http://pkgbuild.sourceforge.net/>
- Write GUI for the package manager, like PackageKit
  - <http://www.packagekit.org/>
- Implement the following:
  - Ability to directly install into a VMware disk image.
  - Ability to boot from a loopback file on another filesystem like NTFS
- Make the new packaging crash-proof



# Notes

- Refer to handout for additional notes and guides.

# open



USE



IMPROVE



EVANGELIZ  
E

## Thank you!

Moinak Ghosh

Moinak.Ghosh@Sun.COM

<http://blogs.sun.com/moinakg/>

“open” artwork and icons by chandan:

<http://blogs.sun.com/chandan>

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