

open



USE



IMPROVE



EVANGELIZE

Distribution Constructor: A Hands on Workshop

Anil Gulecha,
Student, JSSATE & CA Tech Lead, Asia

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



DC : An Introduction

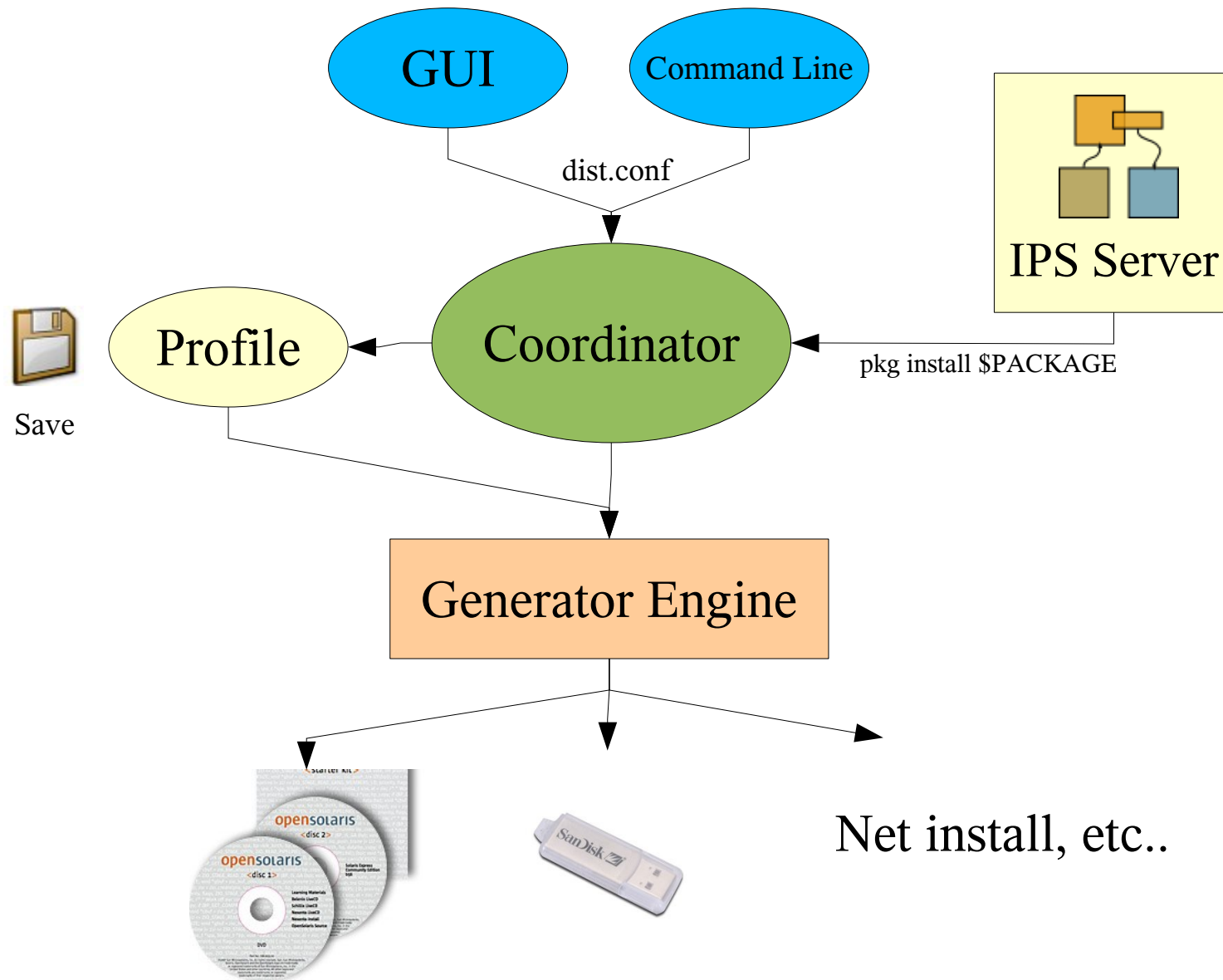
- OpenSolaris is a young and evolving OS.
- Learning from Linux's mistakes.
- Being built with multiple distributions in mind.
- Features everyone uses (LiveCD, packaging, Installer ..)
- Prebuilt packages
- Framework for distributions
- Takes care of the boring work



Features of your Distro

- It will be Live.
- Your choice of Desktop and software.
- It will be optimized with fast CD I/O.
- It can be used from a CD, DVD, USB key, Memory Cards, Music Players, Cell phones (and Hard disk :)
- Can use ZFS when installed to Hard Disk

DC Design



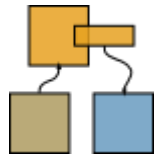


Image Packaging System

- Brand new repository based packaging system
 - *pkg install/uninstall pidgin*
- Allows installation to images (partial/full)
 - *pkg image-create -F -a example.com=http://pkg.example.com:10000 ..*
- Gets packages from IPS server
- Package is a collection of files and actions.
 - Types of actions: file, directory, (hard) link, driver
- Dependency checking built in.



Image Packaging System (cont'd)

- Package Nomenclature:
- 3/4 characters symbolize source, followed by package name.
 - SUNWPython : Python package, built by Sun
 - SFELibntp : libntp library from Spec files Extra
 - CSWfirefox : firefox from Blastwave (CSW)
- `$ pkg list -t dir,file,link,hardlink -o :name,mode,pkg.size,path,target zfs`

-	:NAME	MODE	SIZE	PATH	TARGET
-	dir	0755		etc	
-	dir	0755		etc/fs	
-	dir	0755		etc/fs/zfs	
-	link			etc/fs/zfs/mount	../../../../sbin/zfs
-	link			etc/fs/zfs/umount	../../../../sbin/zfs
-	..				

Structure of a Distribution

- Grub is the OpenSolaris boot loader
- Miniroot is a standalone file that is loaded into RAM by grub.
- Miniroot has the minimal bits of the OS required for the system to boot.
- Applications installed in /usr
- /usr forms the biggest portion of the filesystem



Live CD



DC: Initialization

- Setting up various paths in *dist.conf*
 - `DIST_PKG_LIST` (File: list of packages to be included in distro)
 - `DIST_MICROROOT_LIST` (File: List of packages to be included in distribution)
 - `DIST_MICROROOT_MOD` (Additional changes to microroot)
 - `DIST_ISO_SORT` (File: Sorted listing of files on CD)
 - `DIST_PKG_SERVER` (Address of IPS repository)
 - `DIST_PROTO` (Temp directory where packages will be installed)
 - `DIST_ID` (The name of your distro)
 - `DIST_ISO/DIST_USB` (Full path to the image to be generates)
 - `QUIT_ON_PKG_FAILURES` (What to do after a bad package?)



DC: Selecting Packages

- Which packages go into Miniroot
 - SUNWcsd, SUNWcs, SUNWdoc, SUNWcnetr, SUNWckr,,SUNWkvm, SUNWcakr, ..
 - Take the supplied default and modify accordingly
 - Miniroot exists in the RAM, which is a limited resource, so choose wisely.
- All other applications go into distro packages.
 - Desktop Environment, Internet/Office apps, backup/restore utilities, etc



DC: Building

- Currently run *build_dist.bash*
 - GUI will have building features in the future
- The steps are as follows
 - Directories setup/created
 - *pkg_retrieve_init* (The repository is initialized)
 - *pkg_list_verify* (Check for availability of all packages)
 - *pkg_retrieve* (All installed to the proto area)
 - Repeat for miniroot
- Build ISO
 - ***mkisofs*** *-o \$DIST_ISO -b boot/grub/stage2_eltorito -c .catalog*



DC: A little testing

- Burn generated iso on a CD/DVD and test
- Bootup your distro, and check that
 - Your desktop behaves as required
 - All applications specific to your distro are installed
 - Environment variables are set as required
 - GNU commands?
 - Opensolaris commands?
 - USB install works? Boots?
- Additional changes

DC: Optimizations

- Boot into your freshly created distribution
 - You'll boot into a live environment in 5-10 minutes
- Optimizing bootup
 - Boot your distro using “*Solaris 32 bit I/O tracing*” in Grub
 - Bootup proceeds. A file '*traceout*' saved on hard disk
 - Boot into hard disk (SX) and run *proc_tracedata* on the file
 - File *iso.sort* generated.
 - This contains the reorder of files on CD such that they are accessed linearly when booting up.
 - Rerun DC and have your super fast distro
 - **First time such optimizations have been automated.**



DC: Spread the word

- www.opensolaris.org : The opensolaris community
- Do you want a project page to host your distro?
- What is the goal and who is the target audience?
- Mailing lists, web hosting, brand building: Get the community involved.
- Your ideas?



Want to get involved?

BUILD YOUR DISTRO!

- <http://opensolaris.org/os/project/caiman/>
- <http://opensolaris.org/os/project/pkg>
- caiman-discuss@opensolaris.org
 - *Appliance*
 - *Gaming*
 - *Educational*
 - *Media center*
 - *NAS*
 - *Other..*



Want to get involved? (2)

- Projects you can help with:
 - Reduce the size of miniroot
 - Extend DC to use multiple repositories
 - Build GUI tool that can create branding packages
 - Create/maintain IPS packages
 - Update UDFS on Solaris
 - Get Gparted working fully
 - More..
- <http://opensolaris.org/os/project/caiman/>



Q & A

Anil Gulecha

anil.verve@gmail.com