Dracut: Introduction and Overview

Cong Wang, Software Engineer, Red Hat

May 28, 2012

What's an initial ramdisk

- To assistant Linux OS to boot from the kernel into the init on rootfs
- Does preparations before the real root file system can be mounted
- initrd is a ram based block device
- while initramfs is ramfs based, to replace initrd

What's dracut

Dracut is a generic initramfs infrastructure

- It is aimed to be distro-independent and hardware-independent
- It is highly configurable and flexible
- It is modularized, it is very easy to add your own dracut module
- Its code is much cleaner (for developers)

What does dracut do

- Creating an initramfs to boot the kernel
- Packing all necessary kernel modules for booting the kernel
- Packing all needed utilities to mount the rootfs
- Providing the initial init process
- Mounting specified roofs and switching to the init on rootfs finally
- Mounting non-root filesystems, required by kdump (new)

Running dracut

- Creating an initramfs: `-f`, `-H`, `-l`
- Inspecting an initramfs: `lsinitrd`
- Adding specified kernel modules: `--add-drivers`
- Adding specified utilites: `-I`
- Adding specified dracut modules: `--add`
- Verbose output: `-M`, `-v`, `--debug`

Logs and configure files

Outside of initramfs:

- /var/log/dracut.log, or /tmp/dracut.log, logfile of initramfs image creation
- /etc/dracut.conf, /etc/dracut.conf.d/*.conf, configure file for running dracut, see dracut.conf(5)

Inside of initramfs:

- /etc/conf.d/*, sourced in the initramfs to set initial values, but command line options will override them.
- /etc/cmdline, /etc/cmdline.d/*.conf, contain additional command line options.

Dracut boot cmdline

- Specifying rootfs location, network configuration, or debugging parameters in cmdline
- Rootfs cmdline: root=
- Network cmdline: ip=, ifname=, netroot=
- Debug cmdline: rd.debug, rd.break=
- Lots of device-specific cmdline...

Enhancements for kdump

- A new option `--mount`, to mount non-root devices
- A new cmdline “rd.neednet=1”, to bring up networking interface without netroot=
- A new dracut module, “ssh-client” module
- Vlan support in network module and some fixes for bonding/bridge
- Various fixes in some dracut modules

Writing a dracut module

- Provide module-setup.sh in your own directory
- check(): check the existance of kernel modules, utilities
- depends(): the modules you depend on
- installkernel(): install kernel modules
- install(): install everything else, including hooks
  - *hostonly* flag and *mount_needs* flag

Dracut library functions

- instmods(): install kernel modules
- dracut_install(): install a binary in $PATH
- inst(): install a file into the specified place
- inst_hook(): install a specified hook script
- inst_rules(): install a udev rule file

Examples

- busybox module:
  ```
  check() { type -P busybox >/dev/null || return 1 return 255 }
  ```
- network module:
  ```
  depends() { [ -d /etc/sysconfig/network-scripts/ ] && echo
  ifcfg return 0 }
  ```
- btrfs module:
  ```
  installkernel() { instmods btrfs crc32c }
  ```
- fstab-sys module:
  ```
  install() { [ -f /etc/fstab.sys ] && inst /etc/fstab.sys
  inst_hook pre-pivot 00 “$moddir/mount-sys.sh”
  inst_hook cmdline 00 “$moddir/wait-mount-dev.sh” }
  ```

Dracut modules

- base: the base of all dracut modules, providing the normal init in initramfs
- network: bring up various kinds of NIC, parse ip=, ifname=, netroot=
- fstab-sys: mount non-root devices in /etc/fstab.sys
- kernel-modules: install all necessary kernel modules
- rootfs-block: bring up and mount root device
- plymouth: a graphic bootsplash
- Lots of other device-specific modules. . .

Hooks and breakpoints

In timeline:

- cmdline: in the very beginnnig, interpreting your own cmdlines
- pre-udev: before starting udevd, e.g., writing udev rules
- pre-trigger: after starting udevd, setting udev environment variables
- initqueue: settled/timeout/finished
- pre-mount: right before mounting rootfs
- mount: mount the real root filesystem
- pre-pivot: after mounting rootfs, before switching over to the rootfs init
- cleanup: doing final cleanups before switching to rootfs init

The whole picture

- The bootloader loads the kernel and its initramfs
- Kernel boots and mounts the initramfs
- Kernel runs the init in initramfs
  - Basic setup (Hooks: cmdline, pre-udev)
  - Start Udev (Hooks: pre-trigger)
  - Trigger Udev (Initqueue)
  - Wait for jobs or udev settled (Initqueue settled, finished)
  - Found root device (Hooks: pre-mount, mount, pre-pivot)
  - Cleanup and switch to the new init (Hooks: cleanup)
- Systemd will be in charge of the rest boot process

The new mkdumprd

- Uses a “kdumpbase” module
- Installs makedumpfile etc. in install()
- Parses kdump.conf into boot cmdline and write them into /etc/cmdline.d/
- Parses some kdump.conf options into dracut parameters
- Uses `--mount` to mount all devices in the dump targets
- Uses a pre-pivot hook to do the kernel dump finally

TODO

- Add module-specific parameters, e.g. `--sshkey`
- Remove `rd.neednet=1`
- Support complex network configurations, e.g. vlan tagged bonding
- Documentation and code cleanups

References

- http://fedoraproject.org/wiki/Dracut
- https://dracut.wiki.kernel.org/index.php/Main_Page
- http://www.harald-hoyer.de/personal/blog/dracut-talk-on-fosdem-2010
- http://www.kernel.org/pub/linux/utils/boot/dracut/dracut.html