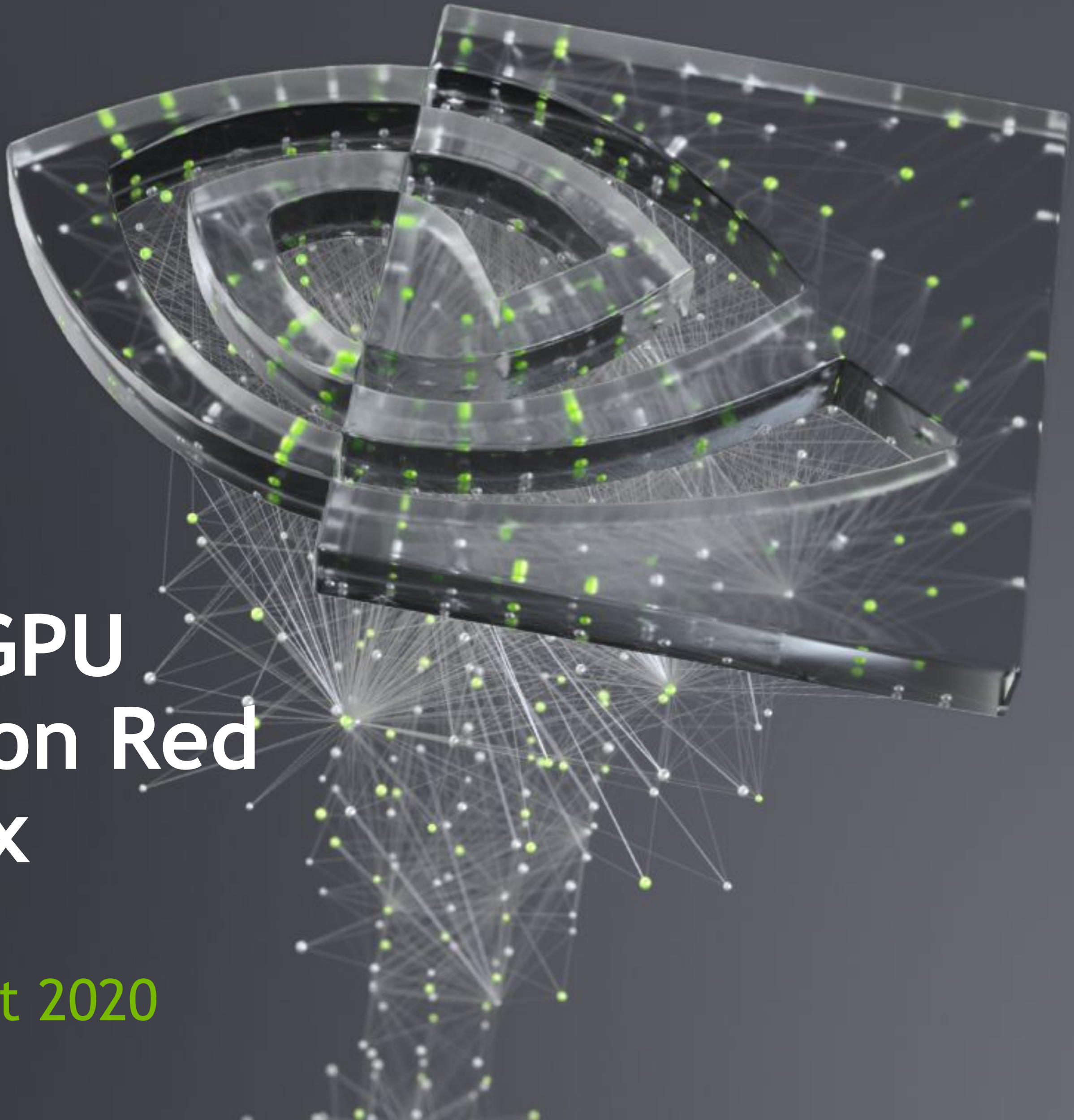


NVIDIA.

Simplifying NVIDIA GPU Driver Deployment on Red Hat Enterprise Linux

Kevin Mittman, Red Hat Summit 2020

AGENDA

- Precompiled kmod packages
- RHEL 7.x lessons
- RHEL 8.x tech preview
- RPM .spec files on GitHub
- Demo
- Questions & Answers

Precompiled kmod packages

What are they?

Terminology as used in this presentation

- kmod: Linux kernel module, a set of loadable drivers
- DKMS: mechanism to re-compile out-of-tree modules on kernel update
- precompiled: pre-built NVIDIA drivers for a kernel (without linking)
- package: DEB, RPM, etc. file archive with pre/post install scriptlets
- package manager: apt/dnf/yum/zypper utility to install packages
- transitive closure: install or remove all packages in stream as one unit
- branch: driver builds from the same major version (ex: 418 or 440)

Precompiled kmod packages

Why would I want them over DKMS?

Benefits

- Removes *gcc* dependency ⇒ no compiler installation required
- Removes *dkms* dependency ⇒ EPEL repository not required
- Removes *kernel-{devel,headers}* deps ⇒ no black screen if missing*
- Pre-compiled ⇒ Faster boot up after driver and/or kernel updates
- Pre-tested ⇒ Kernel and driver combination has been validated

* Mismatched or forgetting to *yum/dnf install kernel-devel-$(uname -r) kernel-headers-$(uname -r)* is the most common NVIDIA driver installation issue. WIth the nouveau driver blacklisted, this can lead to Xorg display server unable to load.

Precompiled kmod packages

Why would I NOT want them over DKMS?

Limitations

- Only official RHEL kernels supported by NVIDIA (no custom kernels[1])
- Driver version and kernel version string must match exactly
- Reliant on kmod package availability for each kernel update[2]

[1] Instructions for building precompiled packages for custom kernels using the .spec files on GitHub is discussed later in this presentation
[2] To avoid system breakage, plugin for package manager will prevent install of kernel updates until compatible kmod package available

Precompiled kmod packages

How does it work?

Building kmod package

1. Compile .o files for NVIDIA kernel modules targeting a specific kernel.
2. Link the .o files against the kernel version string to build the .ko files
3. Sign .ko with X.509 certificate, detach the signature & delete the .ko[1]
4. Ship the .o files and detached signatures in the resulting RPM package

Installing kmod package

1. Post-install script links the packaged .o files to reproduce the .ko files
2. Re-attach signature to sign[2] .ko files; verifies they match

[1] Distributing proprietary binaries pre-linked against the Linux kernel would be a GPL violation
[2] If certificate trusted, this would allow for UEFI Secure Boot support; currently not trusted.

Precompiled kmod packages

Production plan

Enablement matrix

- The most recent NVIDIA driver build, per supported non-EOL branch (see Tesla lifecycle policy for details)
- The most recent kernel of the most recent RHEL 8 minor release
- For x86_64 architecture
- Precompiled kmod packages provided for each kernel update
- ETA is to be determined

RHEL 7.x lessons

Streams: fake it ‘till you make it

Implementation: x3 copies of each driver package per version

flavor	*latest*	*branch-418*	*latest-dkms*
stream	precompiled, ToT	precompiled, locked @ 418.x	legacy DKMS, ToT

kmod-nvidia-${flavor}(-%{kernel}.${driver})	nvidia-driver-${flavor}-NVML
nvidia-driver-${flavor}	nvidia-libXNVCtrl-${flavor}
nvidia-driver-${flavor}-cuda	nvidia-libXNVCtrl-${flavor}-devel
nvidia-driver-${flavor}-cuda-libs	nvidia-modprobe-${flavor}
nvidia-driver-${flavor}-devel	nvidia-persistenced-${flavor}
nvidia-driver-${flavor}-libs	nvidia-settings-${flavor}
nvidia-driver-${flavor}-NvFBCOpenGL	nvidia-xconfig-${flavor}

RHEL 7.x lessons

3 sets of packages, name scheme specially crafted

cuda-compat
cuda-drivers
yum-plugin-nvidia

Requires yum plugin to filter

kmod-latest-dkms	kmod-nvidia-latest-%{kernel}.r418.xx	kmod-nvidia-branch-418-%{kernel}.r418.xx
nvidia-driver-latest-dkms	nvidia-driver-latest	nvidia-driver-branch-418
nvidia-driver-latest-dkms-NVML	nvidia-driver-latest-NVML	nvidia-driver-branch-418-NVML
nvidia-driver-latest-dkms-NvFBCOpenGL	nvidia-driver-latest-NvFBCOpenGL	nvidia-driver-branch-418-NvFBCOpenGL
nvidia-driver-latest-dkms-cuda	nvidia-driver-latest-cuda	nvidia-driver-branch-418-cuda
nvidia-driver-latest-dkms-cuda-libs	nvidia-driver-latest-cuda-libs	nvidia-driver-branch-418-cuda-libs
nvidia-driver-latest-dkms-devel	nvidia-driver-latest-devel	nvidia-driver-branch-418-devel
nvidia-driver-latest-dkms-libs	nvidia-driver-latest-libs	nvidia-driver-branch-418-libs
nvidia-libXNVCtrl-latest-dkms	nvidia-libXNVCtrl-latest	nvidia-libXNVCtrl-branch-418
nvidia-libXNVCtrl-latest-dkms-devel	nvidia-libXNVCtrl-latest-devel	nvidia-libXNVCtrl-branch-418-devel
nvidia-modprobe-latest-dkms	nvidia-modprobe-latest	nvidia-modprobe-branch-418
nvidia-persistenced-latest-dkms	nvidia-persistenced-latest	nvidia-persistenced-branch-418
nvidia-settings-latest-dkms	nvidia-settings-latest	nvidia-settings-branch-418
nvidia-xconfig-latest-dkms	nvidia-xconfig-latest	nvidia-xconfig-branch-418
DKMS	Precompiled	Precompiled, stay on branch 418

RHEL 7.x lessons

A rocky start

June 2019

- Launched RHEL7 precompiled tech preview repository (defunct)
- CI/CD pipelines were not ready to keep up with kernel updates

August 2019

- CUDA 10.1 update 2 (418.87.00) released with the packaging changes
- Due to transitive closure (for yum swap), driver install all or nothing
- nvidia-settings, a GTK-based application, has implicit dependencies on several graphical libraries; undesirable on headless servers

RHEL 7.x lessons

Two weeks later

- Reverted nvidia-settings, nvidia-libXNVCtrl, nvidia-libXNVCtrl-devel back to non-"stream" packages

- No longer part of a stream, thus need to install cuda-drivers metapackage post-install to pull in above non-"stream" packages

- As fallout from sudden reversal of Provides/Obsoletes/Conflicts, dependency hell in resolve, preventing explicit version install of nvidia-settings in CUDA repo

8

RHEL 8.x tech preview

Modularity streams to the rescue

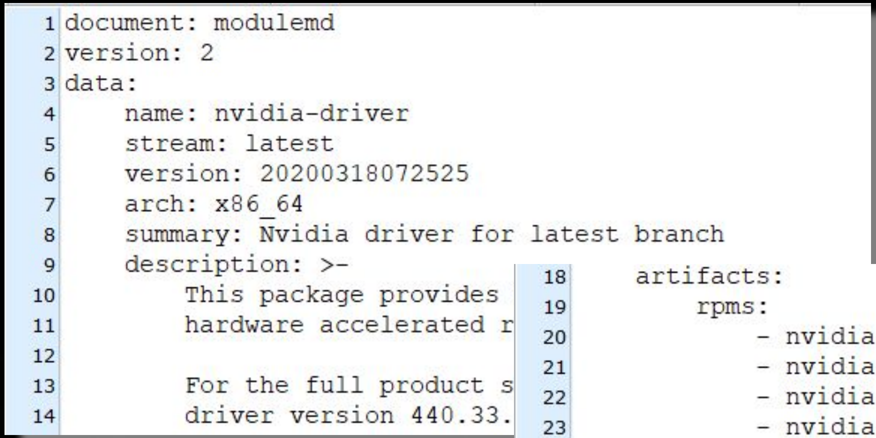
Implementation: modules.yaml

flavor	*latest*	*440*	*latest-dkms*
stream	precompiled, ToT	precompiled, locked @ 440.x	legacy DKMS, ToT

kmod-nvidia-${driver}-%{kernel}-${driver}	nvidia-driver-NVML
kmod-nvidia-latest-dkms	nvidia-kmod-common
nvidia-driver	nvidia-libXNVCtrl
nvidia-driver-cuda	nvidia-libXNVCtrl-devel
nvidia-driver-cuda-libs	nvidia-modprobe
nvidia-driver-devel	nvidia-persistenced
nvidia-driver-libs	nvidia-settings
nvidia-driver-NvFBCOpenGL	nvidia-xconfig

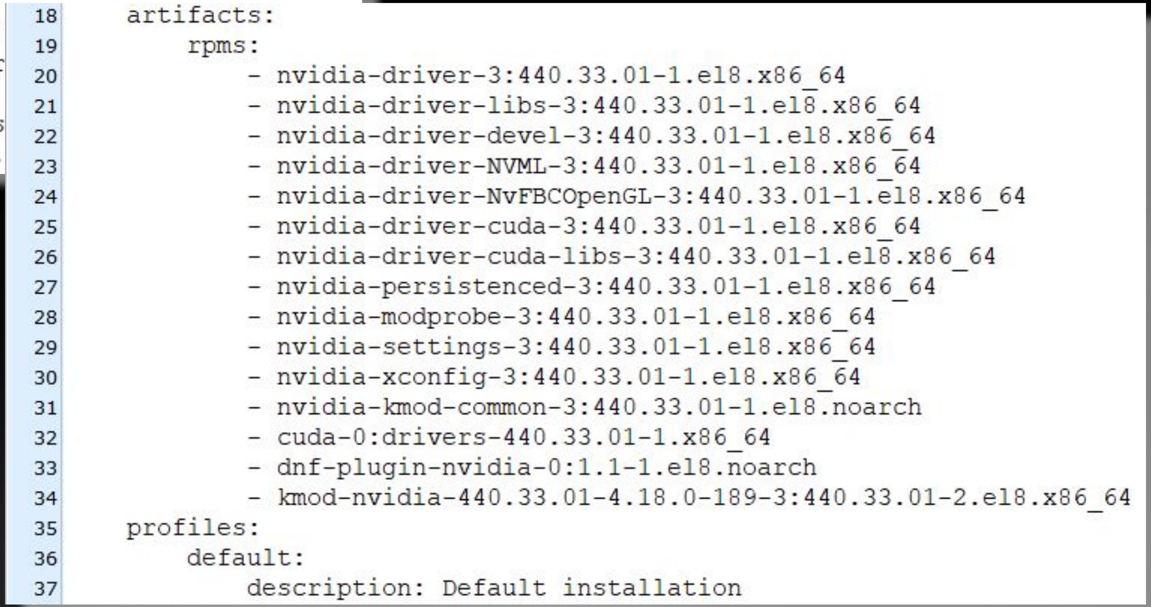
RHEL 8.x tech preview

modules.yaml



```
document: modulemd
version: 2
data:
    name: nvidia-driver
    stream: latest
    version: 20200318072525
    arch: x86_64
    summary: Nvidia driver for latest branch
    description: >-
        This package provides
        hardware accelerated r

        For the full product s
        driver version 440.33.
```



```
    artifacts:
        rpms:
            - nvidia-driver-3:440.33.01-1.el8.x86_64
            - nvidia-driver-libs-3:440.33.01-1.el8.x86_64
            - nvidia-driver-devel-3:440.33.01-1.el8.x86_64
            - nvidia-driver-NVML-3:440.33.01-1.el8.x86_64
            - nvidia-driver-NvFBCOpenGL-3:440.33.01-1.el8.x86_64
            - nvidia-driver-cuda-3:440.33.01-1.el8.x86_64
            - nvidia-driver-cuda-libs-3:440.33.01-1.el8.x86_64
            - nvidia-persistenced-3:440.33.01-1.el8.x86_64
            - nvidia-modprobe-3:440.33.01-1.el8.x86_64
            - nvidia-settings-3:440.33.01-1.el8.x86_64
            - nvidia-xconfig-3:440.33.01-1.el8.x86_64
            - nvidia-kmod-common-3:440.33.01-1.el8.noarch
            - cuda-0:drivers-440.33.01-1.x86_64
            - dnf-plugin-nvidia-0:1.1-1.el8.noarch
            - kmod-nvidia-440.33.01-4.18.0-189-3:440.33.01-2.el8.x86_64
    profiles:
        default:
            description: Default installation
```

1. $ createrepo_c -v --database .
2. $ ./genmodules.py . modules.yaml
3. $ modifyrepo_c modules.yaml ./repodata

RHEL 8.x tech preview

Now available!

Join the RHEL8 precompiled tech preview repository* !

```
$ sudo dnf config-manager \
   --add-repo=https://developer.download.nvidia.com/compute/cuda
   /preview/repos/rhel8/x86_64/techpreview_nvidia_rh_drv.repo

$ sudo dnf module install nvidia-driver:latest
```

* https://developer.download.nvidia.com/compute/cuda/preview/repos/rhel8/x86_64/README.html

xkcd.com/1597

RPM .spec files on GitHub

git push

Now available

https://github.com/NVIDIA/yum-packaging-precompiled-kmod

Coming soon

Rest of the driver packaging git repos with RPM .spec templates

Contributions welcome

Fork, commit, pull request

RPM .spec files on GitHub

git clone

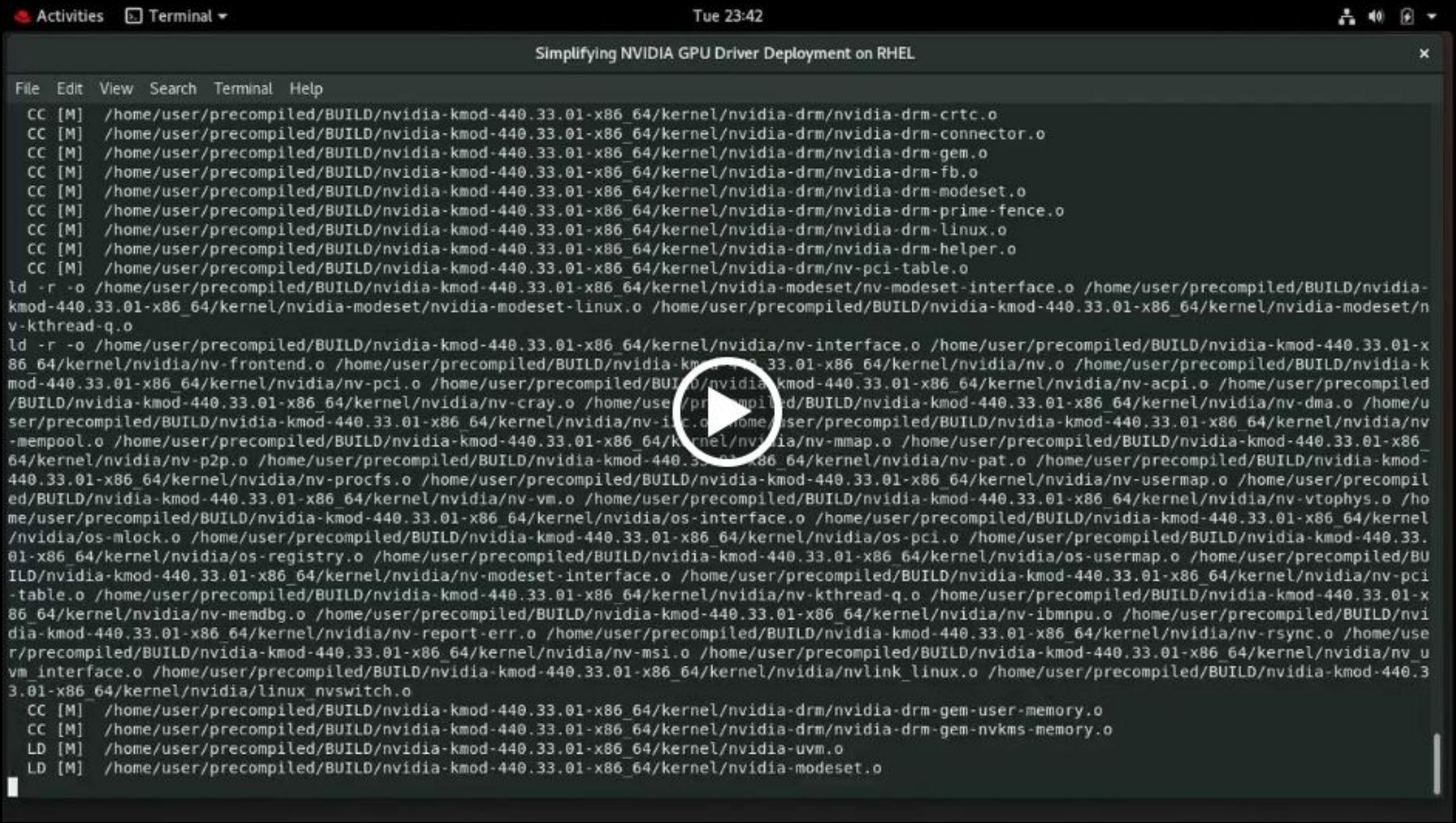
1. Clone https://github.com/nvidia/yum-packaging-precompiled-kmod
2. Generate a X.509 certificate and copy into the repo
3. Build kmod-nvidia.spec with the appropriate parameters[1]

```
$ rpmbuild --define "%_topdir $(pwd)" --define "debug_package %{nil}" \
  --define "kernel $kernel" --define "kernel_release $release" \
  --define "kernel_dist $dist" --define "driver $version" --define "epoch 3" \
  --define "driver_branch $stream" -v -bb SPECS/kmod-nvidia.spec
```

4. Sign the RPM package with GPG key
5. Copy {yum,dnf}-plugin-nvidia from the CUDA repository to RPMS/<arch>
6. Copy the rest of the driver packages (of same version & flavor) to RPMS/<arch>
7. Generate the repodata[2]

[1] stream should be 'latest' for precompiled or to lock to a branch 'XXX' (RHEL8) or 'branch-XXX' (RHEL7)
[2] RHEL8 additionally requires genmodules.py Python script to generate modules.yaml for module stream support

Demo


Activities
Terminal
Tue 23:42
Simplifying NVIDIA GPU Driver Deployment on RHEL
File Edit View Search Terminal Help

```
  CC [M]  /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia-drm/nvidia-drm-crtc.o
  CC [M]  /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia-drm/nvidia-drm-connector.o
  CC [M]  /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia-drm/nvidia-drm-gem.o
  CC [M]  /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia-drm/nvidia-drm-fb.o
  CC [M]  /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia-drm/nvidia-drm-modeset.o
  CC [M]  /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia-drm/nvidia-drm-prime-fence.o
  CC [M]  /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia-drm/nvidia-drm-linux.o
  CC [M]  /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia-drm/nvidia-drm-helper.o
  CC [M]  /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia-drm/nv-pci-table.o
ld -r -o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia-modeset/nv-modeset-interface.o /home/user/precompiled/BUILD/nvidia-
kmod-440.33.01-x86_64/kernel/nvidia-modeset/nvidia-modeset-linux.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia-modeset/n
v-kthread-q.o
ld -r -o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv-interface.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x
86_64/kernel/nvidia/nv-frontend.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv.o /home/user/precompiled/BUILD/nvidia-k
mod-440.33.01-x86_64/kernel/nvidia/nv-pci.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv-acpi.o /home/user/precompiled
/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv-cray.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv-dma.o /home/u
ser/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv-i2c.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv
-mempool.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv-mmap.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_
64/kernel/nvidia/nv-p2p.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv-pat.o /home/user/precompiled/BUILD/nvidia-kmod-
440.33.01-x86_64/kernel/nvidia/nv-procfs.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv-usermap.o /home/user/precompil
ed/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv-vm.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv-vtophys.o /ho
me/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/os-interface.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel
/nvidia/os-mlock.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/os-pci.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.
01-x86_64/kernel/nvidia/os-registry.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/os-usermap.o /home/user/precompiled/BU
ILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv-modeset-interface.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv-pci
-table.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv-kthread-q.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x
86_64/kernel/nvidia/nv-memdbg.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv-ibmnpu.o /home/user/precompiled/BUILD/nvi
dia-kmod-440.33.01-x86_64/kernel/nvidia/nv-report-err.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv-rsync.o /home/use
r/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv-msi.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nv_u
vm_interface.o /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia/nvlink_linux.o /home/user/precompiled/BUILD/nvidia-kmod-440.3
3.01-x86_64/kernel/nvidia/linux_nvswitch.o
  CC [M]  /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia-drm/nvidia-drm-gem-user-memory.o
  CC [M]  /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia-drm/nvidia-drm-gem-nvkms-memory.o
  LD [M]  /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia-uvm.o
  LD [M]  /home/user/precompiled/BUILD/nvidia-kmod-440.33.01-x86_64/kernel/nvidia-modeset.o
```

Terminal recording (asciinema)

Questions?

compute_installer@nvidia.com

Special Thanks

Akshay Taneja

Karthikeyan Somasundaram

Harmandeep Singh

Samhita Jayasimha

Timm Bäder

Torvald Riegel