如何减少jetson平台的rootfs磁盘使用量

瑞泰新时代（北京）科技有限公司
电话：+86　010-84284669 / 84280996 / 84278927
邮箱：info@realtimes.cn
网址：http://www.realtimes.cn
地址：北京市朝阳区和平西街和平西苑20号楼B座901

手册更新历史

文档版本	更新日期	更新内容	创建人
V1.0	2020-10	创建文档	项目部-36

本指南提供了 jetson 平台减少磁盘使用量的操作说明。

1. 注意事项

为防止不同版本间的差异带来的操作失败等问题，请在操作前将原有系统进行镜像备份。具体操作可参看《XXX 系统备份与恢复》。
注：上述 xxx 为使用的 jetson 的具体型号，例如：TX2、Xavier 等

2. 操作说明

Jetson 平台文件系统基于 ubuntu 示例文件系统。添加了一些与 Nvidia 平台相关的二进制文件（与 BSP 相关)，以支持平台的基本功能。Jetson SDK 组件包括 CUDA，cuDNN，TensorRT，OpenCV，VisionWorks，DeepStream 和多媒体 API 提供了与 AI 相关的支持。

该页面细分了 Nvidia 平台相关的二进制文件和 Jetson SDK 组件，因此用户可以删除未使用的文件以节省空间。还提供了一个按大小列出所有已安装 ubuntu 软件包的命令，供用户根据需要删除未使用的软件包。

2.1 Nvidia 平台相关的二进制文件分布

以下是与 Nvidia 平台相关的二进制文件的分布。

```
/
 ├── boot (43M)              Image(kernel binary), dtb and extlinux.conf. All are not used by Xavier,
Image and extlinux.conf are used by Nano, TX1 and TX2
 ├── etc (536K)              Nvidia specified config file. DO NOT touch
 ├── lib
 │   ├── firmware (4.3M)   Firmwares used by platform. DO NOT touch
 │   ├── modules (47M)       Modules(.ko) used by platform. Can delete some based on needs
 │   └── systemd (32K)      Nvidia specified config file. DO NOT touch
 ├── opt
 │   └── nvidia   (17M)      Jetson gpio python lib and usb device mode service. Can delete based on
needs
 ├── usr
 │   ├── bin (356K)         Nvidia speficied tools. DO NOT touch
 │   ├── lib (130M)         Nvidia speficied libs. DO NOT touch
 │   ├── sbin (2M)          Nvidia speficied tools. DO NOT touch
 │   ├── share (2.5M)      Nvidia specified boot logo, license, icon, etc. DO NOT touch
 │   └── src (234M)         Linux header files and grahics demos. Can delete based on needs
 └── var
      └── nvidia (188K)     Placeholder for nvcam. Can delete based on needs
```

2.2 Jetson SDK 组件

sdkmanager（或较早版本的 jetpack 工具）安装了基于 Jetson OS 的 Jetson SDK 组件，包括 CUDA，cuDNN，TensorRT，OpenCV，VisionWorks，MultiMedia API 和 DeepStream。MultiMedia API 由 tbz2 软件包安装，而其他 API 由 deb 软件包安装。

Jetson SDK 库（静态库和动态库）和示例代码都已安装，可以删除示例代码以节省空间。如果用户的应用程序动态链接到这些库，则可以删除所有静态库。

安装 Jetson SDK 组件后，运行以下命令可以删除所有 deb 文件。

```
sudo apt clean       # clean debs in /var/cache/apt/archives
sudo rm /etc/apt/sources.list.d/*              # remove /var/cudaxxxx, /var/visonworksxxxx from apt source list
sudo rm /var/cuda-repo-10-0-local-10.0.326/ /var/visionworks-repo/ /var/visionworks-sfm-repo/
/var/visionworks-tracking-repo/ -rf        # remove nvidia's debs
```

下表列出了所有 Jetson SDK 组件的简要分类。

```
CUDA
```

Elememt	Path	Size
sample	/usr/local/cuda/samples	198M
dynamic lib	/usr/local/cuda/targets/aarch64-linux/lib/*.so*	723M
static lib	/usr/local/cuda/targets/aarch64-linux/lib/*.a	898M
doc	/usr/local/cuda/doc	231M

```
cuDNN
```

Elememt	Path	Size
sample	/usr/src/cudnn_samples_v7	11M
dynamic lib	/usr/lib/aarch64-linux-gnu/libcudnn*.so*	367M
static lib	/usr/lib/aarch64-linux-gnu/libcudnn*.a	358M

```
TensorRT
```

Elememt	Path	Size
sample	/usr/src/tensorrt	783M
dynamic lib	/usr/lib/aarch64-linux-gnu/libnvcaffe_parser*.so* /usr/lib/aarch64-linux-gnu/libnvinfer*.so* /usr/lib/aarch64-linux-gnu/libnvonnxparser*.so* /usr/lib/aarch64-linux-gnu/libnvparsers*.so*	156M

static lib	/usr/lib/aarch64-linux-gnu/libnvcaffe_parser*.a /usr/lib/aarch64-linux-gnu/libnvinfer*.a /usr/lib/aarch64-linux-gnu/libnvonnxparser*.a /usr/lib/aarch64-linux-gnu/libnvparsers*.a	172M
doc	/usr/share/doc/libnvinfer*	24K

OpenCV

Elememt	Path	Size
sample	/usr/share/OpenCV	11M
dynamic lib	/usr/lib/libopencv*.so*	19M

VisionWorks

Elememt	Path	Size
sample	/usr/share/visionworks* ~/VisionWorks-SFM*Samples	131M
dynamic lib	/usr/lib/libvisionworks*.so*	28M
doc	/usr/share/doc/libvisionworks*	352K

DeepStream

Elememt	Path	Size
sample	/opt/nvidia/deepstream/deepstream*/samples	205M
dynamic lib	/opt/nvidia/deepstream/deepstream*/lib	52M
doc	/opt/nvidia/deepstream/deepstream*/doc	16K

MultiMedia API

Path	Size
/usr/src/tegra_multimedia_api	107M

2.3 删除已安装的 deb 软件包

运行以下命令可以删除所有缓存的 deb 文件

```
sudo apt clean
```

运行以下命令可以阐述所有不需要的已安装软件包

```
sudo apt autoremove
```

运行以下命令可以按大小累出所有安装的 deb 软件包。用户可以手动移除不需要的包，以节省空间。

```
dpkg-query -Wf '${Installed-Size}\t${Package}\n' | sort -rn
```

3. 举例

3.1 删除所有 deb 安装软件包

```
sudo apt clean
sudo rm /etc/apt/sources.list.d/*
sudo rm -rf /var/cuda-repo-10-0-local-10.0.326/ /var/visionworks-repo/ /var/visionworks-sfm-repo/
/var/visionworks-tracking-repo
```

3.2 卸载 ubuntu 桌面和组件

```
sudo apt-get purge gnome-shell
sudo apt-get purge docker*
sudo apt-get purge chromium-browser thunderbird fonts-noto-cjk    libreoffice-common    containerd    snapd
sudo apt-get remove --purge python* libpython*
sudo apt-get purge libwebkit2gtk-4.0-37    libqt5webkit5    ubuntu-wallpapers-bionic freepats
libreoffice-writer libicu60    libreoffice-calc vim-runtime libflite1    libperl5.26 libmozjs-52-0
humanity-icon-theme samba-libs kwin-data perl-modules-5.26 light-themes libjavascriptcoregtk-4.0-18
chromium-browser-l10n
sudo apt-get purge libvisionworks libvisionworks-sfm-dev libvisionworks-sfm-repo
sudo apt-get autoremove
```

3.3 删除一些源代码和示例代码

```
cd /usr/src && sudo rm -rf cudnn_samples_v7 linux-headers-4.9.140-tegra-ubuntu18.04_aarch64
linux-headers-4.9.140-tegra-linux_x86_64    tensorrt nvidia
rm -rf /home/nvidia/VisionWorks-SFM-0.90-Samples
```

4. 如何修复缩减磁盘占用可能带来的库缺失问题。

有时，在删除 ubuntu 软件包后，由于缺少依赖的库，应用程序执行可能会失败。

例如，假设通过以上示例删除软件包后，Jetson Multimedia API 示例无法像下面那样运行。（这只是一个例子，Jetson Multimedia API 经过上述缩减后可以很好地工作。）

```
nvidia@nvidia-desktop:~/jetson_multimedia_api/samples/00_video_decode$ ./video_decode
 ./video_decode: error while loading shared libraries: libv4l2.so.0: cannot open shared object file: No such
file or directory
```

“ ldd”命令显示 libv4l2.so.0 和 libnvjpeg.so 都丢失，如下所示。因此，我们需要从一个干净的平台复制这些库。

```
nvidia@nvidia-desktop:~/jetson_multimedia_api/samples/00_video_decode$ ldd video_decode
        linux-vdso.so.1 (0x0000007f869e8000)
        libpthread.so.0 => /lib/aarch64-linux-gnu/libpthread.so.0 (0x0000007f86916000)
        libv4l2.so.0 => not found
        libEGL.so.1 => /usr/lib/aarch64-linux-gnu/libEGL.so.1 (0x0000007f868f5000)
        libGLESv2.so.2 => /usr/lib/aarch64-linux-gnu/libGLESv2.so.2 (0x0000007f868bf000)
        libX11.so.6 => /usr/lib/aarch64-linux-gnu/libX11.so.6 (0x0000007f86796000)
        libnvbuf_utils.so.1.0.0 => /usr/lib/aarch64-linux-gnu/tegra/libnvbuf_utils.so.1.0.0
(0x0000007f8677c000)
        libnvjpeg.so => not found
        libdrm.so.2 => /usr/lib/aarch64-linux-gnu/libdrm.so.2 (0x0000007f8674a000)
        libstdc++.so.6 => /usr/lib/aarch64-linux-gnu/libstdc++.so.6 (0x0000007f865b7000)
        libgcc_s.so.1 => /lib/aarch64-linux-gnu/libgcc_s.so.1 (0x0000007f86593000)
        libc.so.6 => /lib/aarch64-linux-gnu/libc.so.6 (0x0000007f8643a000)
        /lib/ld-linux-aarch64.so.1 (0x0000007f869bd000)
        libGLdispatch.so.0 => /usr/lib/aarch64-linux-gnu/libGLdispatch.so.0 (0x0000007f8630e000)
        libdl.so.2 => /lib/aarch64-linux-gnu/libdl.so.2 (0x0000007f862f9000)
        libxcb.so.1 => /usr/lib/aarch64-linux-gnu/libxcb.so.1 (0x0000007f862c9000)
        libnvrm.so => /usr/lib/aarch64-linux-gnu/tegra/libnvrm.so (0x0000007f86287000)
        libnvrm_graphics.so => /usr/lib/aarch64-linux-gnu/tegra/libnvrm_graphics.so
(0x0000007f86268000)
        libnvddk_vic.so => /usr/lib/aarch64-linux-gnu/tegra/libnvddk_vic.so (0x0000007f8624a000)
        libnvbuf_fdmap.so.1.0.0 => /usr/lib/aarch64-linux-gnu/tegra/libnvbuf_fdmap.so.1.0.0
(0x0000007f86237000)
        libnvll.so => /usr/lib/aarch64-linux-gnu/tegra/libnvll.so (0x0000007f8621b000)
        libm.so.6 => /lib/aarch64-linux-gnu/libm.so.6 (0x0000007f86161000)
        libXau.so.6 => /usr/lib/aarch64-linux-gnu/libXau.so.6 (0x0000007f8614e000)
        libXdmcp.so.6 => /usr/lib/aarch64-linux-gnu/libXdmcp.so.6 (0x0000007f86139000)
        libnvos.so => /usr/lib/aarch64-linux-gnu/tegra/libnvos.so (0x0000007f8611b000)
        libnvdc.so => /usr/lib/aarch64-linux-gnu/tegra/libnvdc.so (0x0000007f860fc000)
        libbsd.so.0 => /lib/aarch64-linux-gnu/libbsd.so.0 (0x0000007f860da000)
        librt.so.1 => /lib/aarch64-linux-gnu/librt.so.1 (0x0000007f860c3000)
        libnvimp.so => /usr/lib/aarch64-linux-gnu/tegra/libnvimp.so (0x0000007f860ae000)
```

在一个干净的平台上，我们可以使用“ find”和“ ls”命令来获取所有必需的库，如下所示。

```
nvidia@nvidia-xavier:~$ find /usr/lib/ -name "libv4l2.so*"
 /usr/lib/aarch64-linux-gnu/libv4l2.so.0
 /usr/lib/aarch64-linux-gnu/libv4l2.so
 /usr/lib/aarch64-linux-gnu/libv4l2.so.0.0.0
 /usr/lib/aarch64-linux-gnu/tegra/libv4l2.so.0
 /usr/lib/aarch64-linux-gnu/libv4l2.so.0.0.999999
```

```
nvidia@nvidia-xavier:/usr/lib/aarch64-linux-gnu$ ls -l /usr/lib/aarch64-linux-gnu/libv4l2.so*
 lrwxrwxrwx 1 root root      12 Dec 26 16:15 /usr/lib/aarch64-linux-gnu/libv4l2.so -> libv4l2.so.0
 lrwxrwxrwx 1 root root      21 Dec 19 14:43 /usr/lib/aarch64-linux-gnu/libv4l2.so.0 ->
libv4l2.so.0.0.999999
 -rw-r--r-- 1 root root 55424 Feb 10  2018 /usr/lib/aarch64-linux-gnu/libv4l2.so.0.0.0
 lrwxrwxrwx 1 root root      18 Dec 10 15:08 /usr/lib/aarch64-linux-gnu/libv4l2.so.0.0.999999 ->
tegra/libnvv4l2.so
```

恢复 libv4l2 后 (/usr/lib/aarch64-linux-gnu/libv4l2.so.0, /usr/lib/aarch64-linux-gnu/libv4l2.so.0.0.0, /usr/lib/aarch64-linux-gnu/libv4l2.so.0.0.999999 and /usr/lib/aarch64-linux-gnu/tegra/libnvv4l2.so) and libnvjpeg(/usr/lib/aarch64-linux-gnu/tegra/libnvjpeg.so)，请记住运行“ sudo ldconfig”更新/etc/ld.so.cache
现在，重新运行 video_decode，它仍然会失败。这是因为新添加的 libv4l2 和 libnvjpeg 依赖于其他缺少的库，如下所示：

```
nvidia@nvidia-desktop:~/jetson_multimedia_api/samples/00_video_decode$ ldd video_decode
        linux-vdso.so.1 (0x0000007f9abff000)
        libpthread.so.0 => /lib/aarch64-linux-gnu/libpthread.so.0 (0x0000007f9ab2d000)
        libv4l2.so.0 => /usr/lib/aarch64-linux-gnu/libv4l2.so.0 (0x0000007f9aa18000)
        libEGL.so.1 => /usr/lib/aarch64-linux-gnu/libEGL.so.1 (0x0000007f9a9f7000)
        libGLESv2.so.2 => /usr/lib/aarch64-linux-gnu/libGLESv2.so.2 (0x0000007f9a9c1000)
        libX11.so.6 => /usr/lib/aarch64-linux-gnu/libX11.so.6 (0x0000007f9a898000)
        libnvbuf_utils.so.1.0.0 => /usr/lib/aarch64-linux-gnu/tegra/libnvbuf_utils.so.1.0.0
(0x0000007f9a87e000)
        libnvjpeg.so => /usr/lib/aarch64-linux-gnu/tegra/libnvjpeg.so (0x0000007f9a826000)
        libdrm.so.2 => /usr/lib/aarch64-linux-gnu/libdrm.so.2 (0x0000007f9a7f4000)
        libstdc++.so.6 => /usr/lib/aarch64-linux-gnu/libstdc++.so.6 (0x0000007f9a661000)
        libgcc_s.so.1 => /lib/aarch64-linux-gnu/libgcc_s.so.1 (0x0000007f9a63d000)
        libc.so.6 => /lib/aarch64-linux-gnu/libc.so.6 (0x0000007f9a4e4000)
        /lib/ld-linux-aarch64.so.1 (0x0000007f9abd4000)
        libv4lconvert.so.0 => not found
        libdl.so.2 => /lib/aarch64-linux-gnu/libdl.so.2 (0x0000007f9a4cf000)
        libGLdispatch.so.0 => /usr/lib/aarch64-linux-gnu/libGLdispatch.so.0 (0x0000007f9a3a3000)
        libxcb.so.1 => /usr/lib/aarch64-linux-gnu/libxcb.so.1 (0x0000007f9a373000)
```

```
	libnvrm.so => /usr/lib/aarch64-linux-gnu/tegra/libnvrm.so (0x0000007f9a331000)
	libnvrm_graphics.so => /usr/lib/aarch64-linux-gnu/tegra/libnvrm_graphics.so (0x0000007f9a312000)
	libnvddk_vic.so => /usr/lib/aarch64-linux-gnu/tegra/libnvddk_vic.so (0x0000007f9a2f4000)
	libnvbuf_fdmap.so.1.0.0 => /usr/lib/aarch64-linux-gnu/tegra/libnvbuf_fdmap.so.1.0.0
(0x0000007f9a2e1000)
	libnvddk_2d_v2.so => /usr/lib/aarch64-linux-gnu/tegra/libnvddk_2d_v2.so (0x0000007f9a2bc000)
	libnvbufsurface.so.1.0.0 => not found
	libnvll.so => /usr/lib/aarch64-linux-gnu/tegra/libnvll.so (0x0000007f9a2a0000)
	libm.so.6 => /lib/aarch64-linux-gnu/libm.so.6 (0x0000007f9a1e6000)
	libXau.so.6 => /usr/lib/aarch64-linux-gnu/libXau.so.6 (0x0000007f9a1d3000)
	libXdmcp.so.6 => /usr/lib/aarch64-linux-gnu/libXdmcp.so.6 (0x0000007f9a1be000)
	libnvos.so => /usr/lib/aarch64-linux-gnu/tegra/libnvos.so (0x0000007f9a1a0000)
	libnvdc.so => /usr/lib/aarch64-linux-gnu/tegra/libnvdc.so (0x0000007f9a181000)
	libbsd.so.0 => /lib/aarch64-linux-gnu/libbsd.so.0 (0x0000007f9a15f000)
	librt.so.1 => /lib/aarch64-linux-gnu/librt.so.1 (0x0000007f9a148000)
	libnvimp.so => /usr/lib/aarch64-linux-gnu/tegra/libnvimp.so (0x0000007f9a133000)
```

与 libv4l2 一样，恢复 libv4lconvert 和 libnvbufsurface 之后，所有依赖项现在都可用。

```
nvidia@nvidia-desktop:~/jetson_multimedia_api/samples/00_video_decode$ ldd video_decode
	linux-vdso.so.1 (0x0000007f8bb5c000)
	libpthread.so.0 => /lib/aarch64-linux-gnu/libpthread.so.0 (0x0000007f8ba8a000)
	libv4l2.so.0 => /usr/lib/aarch64-linux-gnu/libv4l2.so.0 (0x0000007f8b975000)
	libEGL.so.1 => /usr/lib/aarch64-linux-gnu/libEGL.so.1 (0x0000007f8b954000)
	libGLESv2.so.2 => /usr/lib/aarch64-linux-gnu/libGLESv2.so.2 (0x0000007f8b91e000)
	libX11.so.6 => /usr/lib/aarch64-linux-gnu/libX11.so.6 (0x0000007f8b7f5000)
	libnvbuf_utils.so.1.0.0 => /usr/lib/aarch64-linux-gnu/tegra/libnvbuf_utils.so.1.0.0
(0x0000007f8b7db000)
	libnvjpeg.so => /usr/lib/aarch64-linux-gnu/tegra/libnvjpeg.so (0x0000007f8b783000)
	libdrm.so.2 => /usr/lib/aarch64-linux-gnu/libdrm.so.2 (0x0000007f8b751000)
	libstdc++.so.6 => /usr/lib/aarch64-linux-gnu/libstdc++.so.6 (0x0000007f8b5be000)
	libgcc_s.so.1 => /lib/aarch64-linux-gnu/libgcc_s.so.1 (0x0000007f8b59a000)
	libc.so.6 => /lib/aarch64-linux-gnu/libc.so.6 (0x0000007f8b441000)
	/lib/ld-linux-aarch64.so.1 (0x0000007f8bb31000)
	libv4lconvert.so.0 => /usr/lib/aarch64-linux-gnu/libv4lconvert.so.0 (0x0000007f8b3bc000)
	libdl.so.2 => /lib/aarch64-linux-gnu/libdl.so.2 (0x0000007f8b3a7000)
	libGLdispatch.so.0 => /usr/lib/aarch64-linux-gnu/libGLdispatch.so.0 (0x0000007f8b27b000)
	libxcb.so.1 => /usr/lib/aarch64-linux-gnu/libxcb.so.1 (0x0000007f8b24b000)
	libnvrm.so => /usr/lib/aarch64-linux-gnu/tegra/libnvrm.so (0x0000007f8b209000)
	libnvrm_graphics.so => /usr/lib/aarch64-linux-gnu/tegra/libnvrm_graphics.so (0x0000007f8b1ea000)
	libnvddk_vic.so => /usr/lib/aarch64-linux-gnu/tegra/libnvddk_vic.so (0x0000007f8b1cc000)
	libnvbuf_fdmap.so.1.0.0 => /usr/lib/aarch64-linux-gnu/tegra/libnvbuf_fdmap.so.1.0.0
(0x0000007f8b1b9000)
	libnvddk_2d_v2.so => /usr/lib/aarch64-linux-gnu/tegra/libnvddk_2d_v2.so (0x0000007f8b194000)
	libnvbufsurface.so.1.0.0 => /usr/lib/aarch64-linux-gnu/tegra/libnvbufsurface.so.1.0.0 (0x0000007f8b116000)
	libnvll.so => /usr/lib/aarch64-linux-gnu/tegra/libnvll.so (0x0000007f8b0fa000)
	libm.so.6 => /lib/aarch64-linux-gnu/libm.so.6 (0x0000007f8b040000)
	librt.so.1 => /lib/aarch64-linux-gnu/librt.so.1 (0x0000007f8b029000)
	libXau.so.6 => /usr/lib/aarch64-linux-gnu/libXau.so.6 (0x0000007f8b016000)
	libXdmcp.so.6 => /usr/lib/aarch64-linux-gnu/libXdmcp.so.6 (0x0000007f8b001000)
	libnvos.so => /usr/lib/aarch64-linux-gnu/tegra/libnvos.so (0x0000007f8afe3000)
	libnvdc.so => /usr/lib/aarch64-linux-gnu/tegra/libnvdc.so (0x0000007f8afc4000)
	libbsd.so.0 => /lib/aarch64-linux-gnu/libbsd.so.0 (0x0000007f8afa2000)
	libnvimp.so => /usr/lib/aarch64-linux-gnu/tegra/libnvimp.so (0x0000007f8af8d000)
```