Overview Repositories **Discussions** Projects Packages People

Running autoware rocker on gpu in Jetson Orin or arm64 devices in general #3386

Unanswered **sudhsrik** asked this question in **Q&A**

Category

🙏 Q&A

Labels

None yet

10 participants

sudhsrik on Mar 29, 2023

As per the issue mentioned in the website, https://autowarefoundation.github.io/autoware-documentation/main/installation/autoware/docker-installation/#docker-with-nvidia-gpu-fails-to-start-autoware-on-arm64-devices
I am not able to run the rocker version of autoware on the GPU in my Jetson Orin board. The troubleshooting page suggests making changes to the rocker source code as a stop gap solution until the official fix is available from rocker.

I want to be able to utilize the GPUs in my Jetson Orin board.

Could someone give me an idea of what source code changes I need to make on rocker so that it is able to run on gpu?

↑ 2

5 comments · 20 replies

Oldest Newest Top

amadeuszsz on Mar 29, 2023 Collaborator

You just need to pull rocker repo and build. However, I couldn't run docker container with dev branch as well - rocker tool generates invalid command syntax for Jetson boards (at least in my case). You can pull my changes and force rocker valid syntax:

- change `--nvidia` to `--nvidia runtime`
- add `--group-add video`

↑ 2

7 replies

Show 2 previous replies

sudhsrik on Apr 5, 2023 Author

deviceQuery works now and shows result as pass on docker, but I am seeing errors when I run autoware nodes that require GPU such as lidar_centerpoint. The error message is posted below,

```
root@33d0fea68744:/autoware# ros2 launch lidar_centerpoint
single_inference_lidar_centerpoint.launch.xml pcd_pa
th:=/home/danlaw/autoware_map/sample-map-
planning/pointcloud_map.pcd
detections_path:=test_detections.ply [INFO] [launch]: All log
files can be found below /root/.ros/log/2023-04-05-13-56-30-
701459-33d0fea68744-59484 [INFO] [launch]: Default logging
verbosity is set to INFO [INFO]
[single_inference_lidar_centerpoint_node-1]: process started
with pid [59485] [INFO] [lidar_centerpoint_visualizer.py-2]:
process started with pid [59487]
[lidar_centerpoint_visualizer.py-2] /usr/bin/env: ‘python’: No
such file or directory [ERROR]
[lidar_centerpoint_visualizer.py-2]: process has died [pid
59487, exit code 127, cmd
'/autoware/install/lidar_centerpoint/lib/lidar_centerpoint/lida
r_centerpoint_visualizer.py --ros-args -r
__node:=lidar_centerpoint_visualizer --params-file
/tmp/launch_params_86xk6_t3 --params-file
/tmp/launch_params_amdqj2w0'].
[single_inference_lidar_centerpoint_node-1] terminate called
after throwing an instance of
'thrust::system::detail::bad_alloc'
[single_inference_lidar_centerpoint_node-1]   what():
std::bad_alloc: cudaErrorUnsupportedPtxVersion: the provided
PTX was compiled with an unsupported toolchain. [ERROR]
[single_inference_lidar_centerpoint_node-1]: process has died
[pid 59485, exit code -6, cmd
'/autoware/install/lidar_centerpoint/lib/lidar_centerpoint/sing
le_inference_lidar_centerpoint_node --ros-args -r
__node:=lidar_centerpoint --params-file
/tmp/launch_params_xx5vk2_q --params-file
/tmp/launch_params_i28web0t --params-file
/tmp/launch_params_apxjxqtd --params-file
/tmp/launch_params_utw30f4d --params-file
/tmp/launch_params_dlk5g4pw --params-file
/tmp/launch_params_ze6410l7 --params-file
/tmp/launch_params_wlg6ppp6 --params-file
/tmp/launch_params_jr05j5vv --params-file
/tmp/launch_params_n624x18s --params-file
/tmp/launch_params_ylne7j3d --params-file
/autoware/install/lidar_centerpoint/share/lidar_centerpoint/con
fig/centerpoint_tiny.param.yaml --params-file
/autoware/install/lidar_centerpoint/share/lidar_centerpoint/con
fig/detection_class_remapper.param.yaml --params-file
/tmp/launch_params_89x3bxgh --params-file
/tmp/launch_params_7jsyd42g'].
```

amadeuszsz on Apr 5, 2023 Collaborator

I guess you need an alias. Regarding perception modules, you didn't mention if you are using Jetson Orin AGX or NX. NX for sure will struggle with neural networks due to not sufficient vram. You can monitor GPU usage while running Autoware. As i remember, `nvidia-smi` command won't work, try `sudo tegrastats` to see if there is vram available during Autoware stack execution.

sudhsrik on Apr 5, 2023 Author

I am using Jetson Orin AGX to run Autoware. I tried running `sudo tegrastats` and the planning simulator mentioned in the Autoware tutorials. It doesn't look like Autoware is exhausting the VRAM. But still, since I am running only one single component [lidar_centerpoint], it shouldn't be possible to overload the VRAM, right?

amadeuszsz on Apr 5, 2023 Collaborator

I just noticed PTX error. Even though you fixed alias error, `lidar_centerpoint` ONNX models will not work in its form with Jetson boards and there can be multiple reasons [1][2].

cycyc1949 on Sep 14, 2023 Collaborator

I tried to run it on orin, but it took a long time on the cpu

kaspermeck-arm on Aug 23, 2023 Collaborator

Run lidar centerpoint on Orin

I've been working to get the lidar centerpoint node to work on the Orin as well. Collaborating technically with **@oguzkaganozt** and **@ahuazuipiaoliang**. Find our discussion in the IROS 2023 thread under the *#openad-kit* Discord channel.

@sudhsrik **@amadeuszsz** - sharing my experience here below!

Bring-up steps

1. Clean JetPack 5.1.2 installation using SDK Manager
2. Installed all software (including dev tools) using SDK Manager
3. Update driver from `CUDA 11.4` to `CUDA 11.8`
4. Test `deviceQueryDrv`
5. Test `deviceQueryDrv` inside container
6. Test `deviceQueryDrv` inside Autoware container
7. Run lidar centerpoint inside Autoware container

Step 7 **does not** work.

Logs

CUDA 11.8 driver

```
ubuntu@orin:~/cuda-samples/Samples/1_Utilities/deviceQueryDrv$
LD_LIBRARY_PATH=/usr/local/cuda-11.8/compat:$LD_LIBRARY_PATH
./deviceQueryDrv
```

Note: `LD_LIBRARY_PATH` needs to point to `CUDA 11.8` compat

CUDA 11.8 driver inside container

```
ubuntu@orin:~/cuda-samples/Samples/1_Utilities/deviceQueryDrv$
docker run -it --rm --runtime nvidia --gpus all -v $(pwd):$(pwd)
-w $(pwd) -v /usr/local/cuda-11.8:/usr/local/cuda-11.8 -e
LD_LIBRARY_PATH=/usr/local/cuda-11.8/compat:$LD_LIBRARY_PATH
ubuntu:20.04 /bin/bash
root@cda1b8869f72:/home/ubuntu/cuda-
samples/Samples/1_Utilities/deviceQueryDrv# ./deviceQueryDrv
./deviceQueryDrv Starting...

CUDA Device Query (Driver API) statically linked version
Detected 1 CUDA Capable device(s)

Device 0: "Orin"
  CUDA Driver Version:                           11.8
...
```

Note: set `-e` as explained above, share `/usr/local/cuda-11.8`, set `--runtime nvidia` and `--gpus all`

Lidar centerpoint inside Autoware container

RUN AUTOWARE CONTAINER

```
ubuntu@orin:~$ docker run -it --rm --runtime nvidia --gpus all
/usr/local/cuda-11.8:/usr/local/cuda-11.8 -e
LD_LIBRARY_PATH=/usr/local/cuda-11.8/compat:$LD_LIBRARY_PATH
ghcr.io/autowarefoundation/autoware-universe:humble-latest-
prebuilt-cuda-arm64 /bin/bash
```

(INSIDE CONTAINER) RUN LIDAR CENTERPOINT

```
root@8686fb329f7e:/autoware# ros2 launch lidar_centerpoint
lidar_centerpoint.launch.xml
[INFO] [launch]: All log files can be found below
/root/.ros/log/2023-08-23-14-45-12-092084-8686fb329f7e-77
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [lidar_centerpoint_node-1]: process started with pid [78]
[lidar_centerpoint_node-1] [I] [TRT] [MemUsageChange] Init CUDA:
CPU +221, GPU +0, now: CPU 239, GPU 4351 (MiB)
[lidar_centerpoint_node-1] [I] [TRT] [MemUsageChange] Init CUDA:
CPU +0, GPU +0, now: CPU 258, GPU 4351 (MiB)
[lidar_centerpoint_node-1] [I] [TRT] [MemUsageChange] Init
builder kernel library: CPU +357, GPU +135, now: CPU 615, GPU
```

```
4487 (MiB)
[lidar_centerpoint_node-1] [INFO] [1692802653.034101415]
[lidar_centerpoint]: Using TensorRT FP16 Inference
[lidar_centerpoint_node-1] [I] [TRT] ----------------------------
------------------------------------
[lidar_centerpoint_node-1] [I] [TRT] Input filename:
/autoware/install/lidar_centerpoint/share/lidar_centerpoint/data/pts
[lidar_centerpoint_node-1] [I] [TRT] ONNX IR version:  0.0.6
[lidar_centerpoint_node-1] [I] [TRT] Opset version:    11
[lidar_centerpoint_node-1] [I] [TRT] Producer name:    pytorch
[lidar_centerpoint_node-1] [I] [TRT] Producer version: 1.9
[lidar_centerpoint_node-1] [I] [TRT] Domain:
[lidar_centerpoint_node-1] [I] [TRT] Model version:    0
[lidar_centerpoint_node-1] [I] [TRT] Doc string:
[lidar_centerpoint_node-1] [I] [TRT] ----------------------------
------------------------------------
[lidar_centerpoint_node-1] [W] [TRT] onnx2trt_utils.cpp:369: Your
ONNX model has been generated with INT64 weights, while TensorRT
does not natively support INT64. Attempting to cast down to
INT32.
[lidar_centerpoint_node-1] [INFO] [1692802653.051628950]
[lidar_centerpoint]: Applying optimizations and building TRT CUDA
engine
(/autoware/install/lidar_centerpoint/share/lidar_centerpoint/data/pt
...
[lidar_centerpoint_node-1] [I] [TRT] [MemUsageChange] Init
cuBLAS/cuBLASLt: CPU +212, GPU +715, now: CPU 827, GPU 5202 (MiB)
[lidar_centerpoint_node-1] [I] [TRT] [MemUsageChange] Init cuDNN:
CPU +95, GPU +482, now: CPU 922, GPU 5684 (MiB)
[lidar_centerpoint_node-1] [I] [TRT] Local timing cache in use.
Profiling results in this builder pass will not be stored.
[lidar_centerpoint_node-1] [E] [TRT] 1:
[ltWrapper.cpp::plainGemm::505] Error Code 1: Cublas
(CUBLAS_STATUS_EXECUTION_FAILED)
[lidar_centerpoint_node-1] [E] [TRT] 2:
[builder.cpp::buildSerializedNetwork::636] Error Code 2: Internal
Error (Assertion engine != nullptr failed. )
[lidar_centerpoint_node-1] [ERROR] [1692804430.047830235]
[lidar_centerpoint]: Failed to create serialized network
[lidar_centerpoint_node-1] [ERROR] [1692804430.208273221]
[lidar_centerpoint]: Failed to create context: Engine was not
created
[ERROR] [lidar_centerpoint_node-1]: process has died [pid 78,
exit code -11, cmd
'/autoware/install/lidar_centerpoint/lib/lidar_centerpoint/lidar_cen
--ros-args -r __node:=lidar_centerpoint --params-file
/tmp/launch_params_m45jkl4x --params-file
/tmp/launch_params_tp6wo9sx --params-file
/tmp/launch_params_70g9020r --params-file
/tmp/launch_params_imesky53 --params-file
/tmp/launch_params_36qrk24t --params-file
/tmp/launch_params_ugmojf28 --params-file
/tmp/launch_params_ls7ms3e7 --params-file
/tmp/launch_params_e8tez3s7 --params-file
/tmp/launch_params_bqep4lwr --params-file
/tmp/launch_params_x8akwfk1 --params-file
/autoware/install/lidar_centerpoint/share/lidar_centerpoint/config/c
```

It took a really long time for this to start running before it failed. The lidar centerpoint node works on the AADP + NVIDIA GPU, so it is confirmed that it can work on Arm64, but I was using `CUDA 12.2` driver. This shouldn't matter as Autoware only requires `CUDA 11.6`, the driver is backwards compatible. I have not had time to debug the issue. Any ideas?

↑ 1 1 reply

chishengshih on Aug 24, 2023 Collaborator

Hi,

Here is the alternative to run centerpoint on Orin. The following steps execute ceterpoint_tvm, which uses CPU, and you should expect performance degradation.

There have been discussions on deploying Autoware dockers on nVidia Orin. Rohit from DanLaw tried and succeeded. You may try the instructions provided by Rohit.

1. Setup the base OS (Ubuntu 20 based Jetson Linux) - We are loading this on an SSD so that we can easily swap it out for demo / development versions without many issues.

2. Install dependencies required (docker and rocker installation) - [discussion thread]

3. Use the Autoware official docker file.
   i. While running the docker, we are still having issues with enabling the GPU. The following command from Autoware is used.

   ```
    `rocker -e LIBGL_ALWAYS_SOFTWARE=1 --x11 --user --
   volume $HOME/autoware --
   ghcr.io/autowarefoundation/autoware-universe:latest-
   cuda`
   ```

   ii. Once the docker file is setup, the source code needs to be modified for lidar_centerpoint_tvm needs to be enabled instead of lidar_centerpoint package since GPU is not enabled. The steps to do this can be found at this [discussion thread].

Daniel SHIH

MrOCW on Aug 24, 2023 edited ▾

I am using the arm64 docker image to run everything except lidar_centerpoint, and created a separate JetPack image with the default CUDA11.4 for running the lidar_centerpoint. The topics in rqt seems to flow through correctly

↑ 1 8 replies

Show 3 previous replies

oguzkaganozt on Aug 28, 2023 Maintainer

You're right maybe this can be the problem. Right now Autoware shipped with CUDA 11.6 by default and ORIN only supports 11.4. And we are manually updating CUDA to 12.2 using `cuda-compat` libraries. I will try to build Autoware image with 12.2 CUDA try out on ORIN.

Also as **@kaspermeck-arm** said It should not be about the arm64 architecture, the only thing remains NVIDIA driver differences, CUDA and TensorRT version differences.

👍 1

JonasHablitzel on Sep 14, 2023

Hello,
i am currently running Autoware inside a docker where i build everything from a l4t base container. What i found in the Dockerfile sources for the "normal" Cuda and l4t containers was that the cuda repos differ. So i think the guess from **@MrOCW** holds true for the different Cuda-driver. For reference:

l4t-docker
cuda-docker

```
# in l4t
RUN echo "deb https://repo.download.nvidia.com/jetson/co…… $

# in cuda
echo "deb https://developer.download.nvidia.com/compute/cuda/r
```

so it could maybe be a solution to have the repo as variable in the cuda-ansible playbook and provide there the orin/l4t repo
cuda-ansible-autoware

kaspermeck-arm on Sep 18, 2023 Collaborator

@JonasHablitzel - are you able to run lidar centerpoint inside your l4t base container on Orin targeting the GPU?

JonasHablitzel on Sep 19, 2023

yes i am able to, what was a blocker that i needed to compile PCL by my own (Targeting 1.12.1).

The Performace is not great. In max-settings i get around 4-5FPS. After profiling i found that the main problem is the pointsToVoxels, wich takes around 0,2s for me. The inference time for the centerpoint-tiny NN is around 0.06s, including copy from and to the device.

I'm currently loking into using the shared memory of the ORIN `(cudaHostAlloc(cpuPtr, size, cudaHostAllocMapped))` instead of copying to target, wich could improve the inference time but should have no impact on the pointsToVoxels.

👍 1

kaspermeck-arm on Sep 19, 2023 Collaborator

@JonasHablitzel
From your analysis, can you tell if the `pointsToVoxel` function is issuing SIMD instructions?

Autostone-c on Aug 29, 2023

anybody has fix this ERROR to make lidar_centerpoint_node work on Orin?

↑ 1 1 reply

kaspermeck-arm on Sep 12, 2023 Collaborator

@Autostone-c - unfortunately not. The next release of JetPack with newer default CUDA driver will be necessary. Here's my attempt to summarize the issue

- https://discord.com/channels/953808765935816715/1139166899222093886/1151224135633092689

mitsudome-r on Sep 7, 2023 Maintainer

It's not Orin and it's not lidar_centerpoint, but it seems you can run tensorrt_yolox using Jetson AGX Xavier based ECU by following this instruction

↑ 1 3 replies

oguzkaganozt on Sep 13, 2023 Maintainer

I followed this documentation and it appears that in order to install cuda, cudnn and tensorrt dependencies we need to use Jetson packages rather than standart arm64-sbsa packages for ORIN. Because architecture and GPU bindings seem different.

So as a solution;

- using NVIDIA Jetson Containers as a base(for cuda, cudnn and tensorrt dependencies)
- installing Autoware dependencies
- finally building Autoware from scratch inside the container

This will take hours to validate but I will try to validate the solution.

Autostone-c on Sep 18, 2023

Based on your statement, can autoware(humble-branch) only be run in Orin's Docker environment and cannot be run on the host computer?

oguzkaganozt on Sep 19, 2023 Maintainer

By default, ORIN Jetpack comes with Ubuntu 20.04, ROS-humble requires 22.04 and it also comes with an insufficient CUDA version of 11.4. To tackle those problems I have created a container with Ubuntu 22.04 and tried updating CUDA inside the container by following https://developer.nvidia.com/blog/simplifying-cuda-upgrades-for-nvidia-jetson-users/ but it didn't work.