autowarefoundation / **autoware** Public

Code Issues 42 Pull requests 13 Discussions Actions Security ···

tier4_localization_launch/launch/util/util.launch.py does not start as expected in main branch #4124

New issue

Open 3 tasks done **chivas1000** opened this issue on Jan 26 · 10 comments

chivas1000 commented on Jan 26

Checklist

- ☑ I've read the contribution guidelines.
- ☑ I've searched other issues and no duplicate issues were found.
- ☑ I'm convinced that this is not my fault but a bug.

Description

OS: Ubuntu 22.04 ROS2 humble

steps to reproduce:
launch Autoware.sh
ros2 node list
the corresponding node
"localization/util/crop_box_filter_measurement_range"
"localization/util/voxel_grid_downsample_filter"
"localization/util/random_downsample_filter" are not in the list
but other launch.py with pointcloud preprocessor node such as
"src/sensor_kit/sample_sensor_kit_launch/sample_sensor_kit_launch/launch/pointcloud_preprocessor.launch.py" are in the node list

Expected behavior

the corresponding node
"localization/util/crop_box_filter_measurement_range"
"localization/util/voxel_grid_downsample_filter"
"localization/util/random_downsample_filter" are shown in the ros2 node list

Actual behavior

Assignees

No one assigned

Labels

component:localization type:question

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

5 participants

the corresponding node
"localization/util/crop_box_filter_measurement_range"
"localization/util/voxel_grid_downsample_filter"
"localization/util/random_downsample_filter" are not in the list

Steps to reproduce

launch Autoware.sh
ros2 node list

Versions

OS: Ubuntu 22.04 ROS2 humble Autoware main branch

Possible causes

when I ros2 launch tier4_localization_launch util.py
it saids lacking
"crop_box_filter_measurement_range_param_path" and other parameters
then I set these param to actual path and param and ros2 launch
it started, but no output on the terminal, and no nodes and topic on ros2

Additional context

No response

SakodaShintaro commented on Jan 26 Contributor

Thank you for your report.
I'm sorry, but I don't really understand what `launch Autoware.sh` is. Could you please provide a more detailed or explicit command?

In general, the pointcloud preprocess started by tier4_localization_launch is expected to run in the same process by specifying a container that preprocesses the sensing LiDAR. Therefore, I think an active LiDAR processing container is required.
https://github.com/autowarefoundation/autoware.universe/blob/57cf88d2b85db000ad19ff195eb4106283367edf/launch/tier4_localization_launch/launch/localization.launch.xml#L18

I hope this information helps you.

chivas1000 commented on Jan 26 Author

> Thank you for your report. I'm sorry, but I don't really understand what `launch Autoware.sh` is. Could you please provide a more detailed or explicit command?
>
> In general, the pointcloud preprocess started by tier4_localization_launch is expected to run in the same process by specifying a container that preprocesses the sensing LiDAR. Therefore, I think an active LiDAR processing container is required.
> https://github.com/autowarefoundation/autoware.universe/blob/57cf88d2b85db000ad19ff195eb4106283367edf/launch/tier4_localization_launch/launch/localization.launch.xml#L18
>
> I hope this information helps you.

Thanks for your quick response and sorry for the typo, generally I mean I started Autoware.launch.xml in autoware_launch folder
I will look into the lidar container, is that the ros2 node can show if the lidar container is opened?

SakodaShintaro commented on Jan 26 Contributor

For default setting, localization requires `/sensing/lidar/top/pointcloud_preprocessor/pointcloud_container`.
It is set at localization.launch.xml or passed as arg from tier4_localization_component.launch.xml.
The container can be confirmed by `ros2 node list`

```
~$ ros2 node list | grep /sensing/lidar/top/po lo
/sensing/lidar/top/pointcloud_preprocessor/poi
```

chivas1000 commented on Jan 26 Author

> For default setting, localization requires `/sensing/lidar/top/pointcloud_preprocessor/pointcloud_container`. It is set at localization.launch.xml or passed as arg from tier4_localization_component.launch.xml. The container can be confirmed by `ros2 node list`
>
> ```
> ~$ ros2 node list | grep /sensing/lidar/t i
> /sensing/lidar/top/pointcloud_preprocessor/poin
> ```

thanks for your notice, now I am wondering where would be this container started.
I've noticed that the container you mentioned("/sensing/lidar/top/pointcloud_preprocessor/pointcloud_container") should be at "autoware/src/sensor_kit/sample_sensor_kit_launch/sample_sensor_kit_launch/launch/lidar.launch.xml"

but it seems it didn't started either in my log and ros2 node list:

```
1706259920.6729560 [INFO] [launch]: All log files can be found below /home/cityu/.ros/log/2024-01-26-17-05-20-671971-cityu-Default-string-2967
1706259920.6730738 [INFO] [launch]: Default logging verbosity is set to INFO
1706259965.4922812 [INFO] [launch_ros.actions.load_composable_nodes]: Loaded node '/system/component_state_monitor/component' in container '/system/component_state_monitor/container'
1706259965.6030960 [INFO] [launch_ros.actions.load_composable_nodes]: Loaded node '/system/system_monitor/cpu_monitor' in container '/system/system_monitor/system_monitor/system_monitor_container'
1706259965.6502619 [INFO] [launch_ros.actions.load_composable_nodes]: Loaded node '/system/mrm_comfortable_stop_operator' in container '/system/mrm_comfortable_stop_operator/mrm_comfortable_stop_operator_container'
1706259965.7174656 [INFO] [launch_ros.actions.load_composable_nodes]: Loaded node '/system/mrm_emergency_stop_operator' in container '/system/mrm_emergency_stop_operator/mrm_emergency_stop_operator_container'
1706259966.0107603 [INFO] [launch_ros.actions.load_composable_nodes]: Loaded node '/map/lanelet2_map_loader' in container '/map/map_container'
1706259966.5794673 [INFO] [launch_ros.actions.load_composable_nodes]: Loaded node '/perception/obstacle_segmentation/crop_box_filter' in container 'pointcloud_container'
1706259966.8655772 [INFO] [launch_ros.actions.load_composable_nodes]: Loaded node '/pointcloud_container/glog_component' in container '/pointcloud_container'
1706259967.0621896 [INFO] [launch_ros.actions.load_composable_nodes]: Loaded node '/sensing/lidar/concatenate_data' in container 'pointcloud_container'
```

And I've noticed that autoware.launch.xml launches another pointcloud container "autoware/src/launcher/autoware_launch/autoware_launch/launch/pointcloud_container.launch.py"
and named /pointcloud_container

which one is the container that util.py needed, is that still "/sensing/lidar/top/pointcloud_preprocessor/pointcloud_container"?

SakodaShintaro commented on Jan 26 Contributor

Recently there have been changes about lidar containers.
[autowarefoundation/autoware.universe#6091]

Containers are separated for redundancy.
For Localization, it seems best to connect to the exact container of the point cloud you want to use.

@kminoda
Sorry, I couldn't understand even after looking at the pull request. I think after the change, containers are created

- for each LiDAR process
- one for preprocess such as concat

For example, if there are three LiDARs, top, left, and right, 4 containers are created.

- `/sensing/lidar/top/pointcloud_preprocessor/pointcloud_container`
  - Container for top LiDAR processing
- `/sensing/lidar/left/pointcloud_preprocessor/pointcloud_container`
  - Container for left LiDAR processing
- `/sensing/lidar/right/pointcloud_preprocessor/pointcloud_container`
  - Container for right LiDAR processing
- `/pointcloud_container`
  - Container for concat processing

Is it correct?
In terms of the diagram explained in the details of the [pull request], is the diagram on the left correct?

kminoda commented on Jan 28 Contributor

@SakodaShintaro Sorry for the late reply. Yes, your understanding is correct. In Autoware `use_pointcloud_container` is set true by default, and thus the node diagram on the left is correct.

👍 1

chivas1000 commented on Jan 29 Author

> Recently there have been changes about lidar containers.
> autowarefoundation/autoware.universe#6091
>
> Containers are separated for redundancy. For Localization, it seems best to connect to the exact container of the point cloud you want to use.

@kminoda Sorry, I couldn't understand even after looking at the pull request. I think after the change, containers are created

```
* for each LiDAR process

* one for preprocess such as concat
```

For example, if there are three LiDARs, top, left, and right, 4 containers are created.

```
*
`/sensing/lidar/top/pointcloud_preprocessor/poi

  * Container for top LiDAR processing

*
`/sensing/lidar/left/pointcloud_preprocessor/po

  * Container for left LiDAR processing

*
`/sensing/lidar/right/pointcloud_preprocessor/p

  * Container for right  LiDAR processing

* `/pointcloud_container`

  * Container for concat processing
```

Is it correct? In terms of the diagram explained in the details of the pull request, is the diagram on the left correct?

Dear @SakodaShintaro ,
Sorry I forgot to mention that I've am using the sensor_kit code in galactic manner, in which I've disabled lidar_driver and directly put the pointcloud from rslidar_sdk and remapped as"/sensing/lidar/top/pointcloud" in "autoware/src/universe/autoware.universe/launch/tier4_localization_launch/launch/util/util.launch.py", which worked at galactic branch.

But as the changes which you've mentioned, in main branch, this container(/sensing/lidar/top/pointcloud_preprocessor/pointcloud_container) are created in sensor_kit when launch_driver are set to true. thus the container and all preprocessing node in it didn't launched.

As for this situation, I've looked into nebula_ros package and find there are avaliable driver for robosense helio 32, but lacked **nebula_decoders)/calibration/robosense/Helios5515.csv** thus it cannot be launched. there are no template calibration file for robosense so I am not able to create one from manual.

is it a temporal solution for me to set launch_driver to true, but delete nebula driver related component(DriverRosWrapper and HwInterfaceRosWrapper) and load only the preprocessor component in container, and then remap topic from rslidar_sdk to /sensing/lidar/top/pointcloud_raw_ex?

Again, thanks for your help!

SakodaShintaro commented on Jan 29 Contributor

@chivas1000
Sorry, I misunderstood and thought you were referring to the latest version of Autoware.

(1) Ideal Scenario:

The reason for using a container is to reduce the overhead in topic communication and improve performance. Ideally, you would want 'rslidar_sdk' and 'point cloud preprocessing nodes for NDT' in the same container. The best implementation would be to launch a container and add 'rslidar_sdk' and 'point cloud preprocessing nodes for NDT' to it. Unfortunately, I am not very familiar with device drivers, so I am not sure if this is possible.

(2) Make it work simply:
If you accept the overhead of point cloud copying and just want to run NDT, you should be able to create a new container for point cloud preprocessing.

For example, in my environment, it worked by rewriting `launch/tier4_localization_launch/launch/util/util.launch.py` as follows.

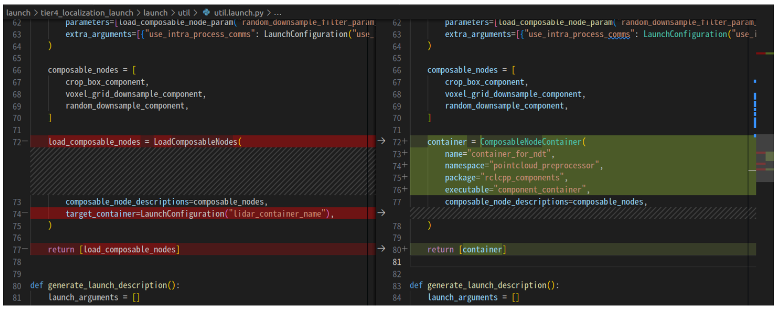
```
~/autoware/src/universe/autoware.universe$ git    f
diff --git a/launch/tier4_localization_launch/launch
index 22a45fe7b8..c2ec70eefc 100644
--- a/launch/tier4_localization_launch/launch/util/u
+++ b/launch/tier4_localization_launch/launch/util/u
@@ -16,7 +16,7 @@ import launch
 from launch.actions import DeclareLaunchArgument
 from launch.actions import OpaqueFunction
 from launch.substitutions import LaunchConfiguratio
-from launch_ros.actions import LoadComposableNodes
+from launch_ros.actions import ComposableNodeContai
 from launch_ros.descriptions import ComposableNode
 import yaml

@@ -69,12 +69,16 @@ def launch_setup(context, *args,
             random_downsample_component,
         ]

-    load_composable_nodes = LoadComposableNodes(
+    container = ComposableNodeContainer(
+        name="container_for_ndt",
+        namespace="pointcloud_preprocessor",
+        package="rclcpp_components",
+        executable="component_container",
         composable_node_descriptions=composable_noc
-        target_container=LaunchConfiguration("lidar
     )

-    return [load_composable_nodes]
+    return [container]


 def generate_launch_description():
```


launch > tier4_localization_launch > launch > util > util.launch.py > ...

```
            parameters=[load_composable_node_param("random_downsample_filter_param_
            extra_arguments=[{"use_intra_process_comms": LaunchConfiguration("use_
        )

    composable_nodes = [
        crop_box_component,
        voxel_grid_downsample_component,
        random_downsample_component,
    ]

    load_composable_nodes = LoadComposableNodes(
        composable_node_descriptions=composable_nodes,
        target_container=LaunchConfiguration("lidar_container_name"),
    )

    return [load_composable_nodes]


def generate_launch_description():
    launch_arguments = []
```

```
            parameters=[load_composable_node_param("random_downsample_filter_param_
            extra_arguments=[{"use_intra_process_comms": LaunchConfiguration("use_i
        )

    composable_nodes = [
        crop_box_component,
        voxel_grid_downsample_component,
        random_downsample_component,
    ]

    container = ComposableNodeContainer(
        name="container_for_ndt",
        namespace="pointcloud_preprocessor",
        package="rclcpp_components",
        executable="component_container",
        composable_node_descriptions=composable_nodes,
    )

    return [container]


def generate_launch_description():
    launch_arguments = []
```

It launchs

```
~$ ros2 node list | grep ndt
/localization/pose_estimator/ndt_scan_matcher
/localization/util/pointcloud_preprocessor/containe
```

Does this suggestion help with your issue? If you have any more questions, please feel free to ask.

idorobotics added the type:question label on Jan 30

SakodaShintaro added the component:localization label on Jan 31

Motsu-san commented on Apr 9

@chivas1000 Is it OK to close this issue?

chivas1000 commented on Apr 10 Author

> @chivas1000 Is it OK to close this issue?

As for now, there is no valid driver to using composable node for rslidar helios-32 to load it into the container(to me it seems nebula driver not working as of 2024.01), only loading pointcloud processing nodes into the container and transfer the pointcloud topic to it(which would add pointcloud copy overhead) works.
It seem to be a tempolarily solution. But have to fix the composable node problem for increasing performance.