Crash during planning if goal is set for the second time #4138

New issue

Closed 3 tasks done **felixf4xu** opened this issue on Jan 31 · 17 comments

felixf4xu commented on Jan 31 • edited

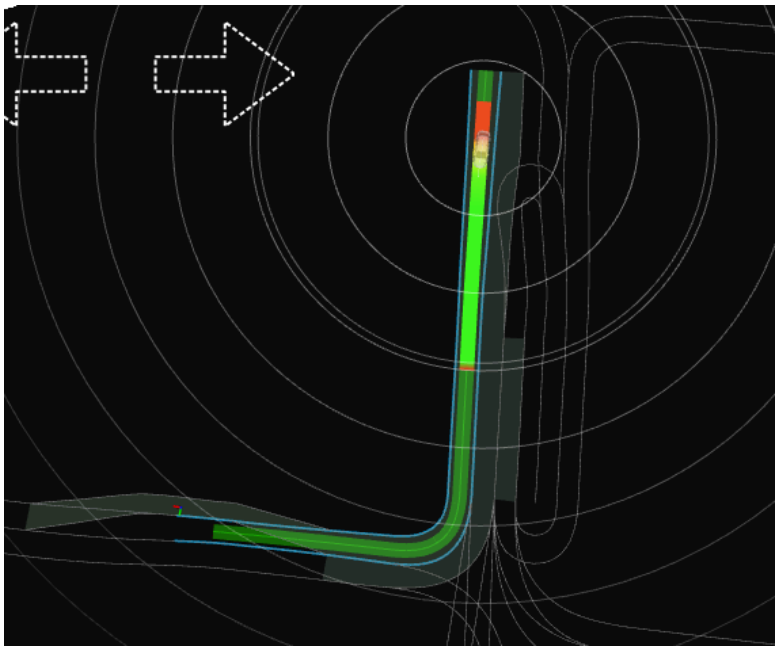
Checklist

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

Description

Hi,

I had a crash during planning (using simple simulator) test. I debugged it for 2 days but had no clue, I'd like to seek any help if possible.



the route with color is the first route, which is ok.
Then I changed the goal position to the next lane (as show in the picture, just one lane upward), there is a crash.

Expected behavior

no crash

Assignees

maxime-clem

Labels

component:planning

meeting:planning-control-wg

Projects

Planning & Control Working Gr...

Status: Done

Milestone

No milestone

Development

No branches or pull requests

3 participants

Actual behavior

```
[component_container_mt-28] *** Aborted at
1706696111 (unix time) try "date -d @170669611…
if you are using GNU date ***
[component_container_mt-28] PC: @
0x0 (unknown)
[component_container_mt-28] *** SIGSEGV (@0x0)
received by PID 3051 (TID 0x7f5591ddf640) from
PID 0; stack trace: ***
[component_container_mt-28]     @
0x7f554ff98046 (unknown)
[component_container_mt-28]     @
0x7f55adc42520 (unknown)
[component_container_mt-28]     @
0x7f55adc97ef4 pthread_mutex_lock
[component_container_mt-28]     @
0x7f55ad5d4014
eprosima::fastdds::dds::detail::ConditionNotifier::a
[component_container_mt-28]     @
0x7f55ad5d46ad
eprosima::fastdds::dds::detail::WaitSetImpl::attach_
[component_container_mt-28]     @
0x7f55adb11525
rmw_fastrtps_shared_cpp::__rmw_wait()
[component_container_mt-28]     @
0x7f55adb68707 rmw_wait
[component_container_mt-28]     @
0x7f55ae2a7848 rcl_wait
[component_container_mt-28]     @
0x7f55ae4266ac rclcpp::Executor::wait_for_work()
[component_container_mt-28]     @
0x7f55ae4293c3
rclcpp::Executor::get_next_executable()
[component_container_mt-28]     @
0x7f55ae430252
rclcpp::executors::MultiThreadedExecutor::run()
[component_container_mt-28]     @
0x7f55ae0e62b3 (unknown)
[component_container_mt-28]     @
0x7f55adc94ac3 (unknown)
[component_container_mt-28]     @
0x7f55add26850 (unknown)
[component_container_mt-28]     @
0x0 (unknown)
```

Steps to reproduce

1. Create a route by setting pose and goal, there should be a planning route displayed.
2. The "Auto" button is ready
3. don't change pose, but change goal position
   -> the "Auto" button is disabled
   -> there is a crash log in the terminal, as showed above.

Versions

os: ubuntu 22
ros2: humber
autoware: main branch

Possible causes

From the crash log, it seems some pointer is access but it is not valid.
The call stack is all for ros/dds, I didn't see my code. I can attach GDB to the node, but from gdb/backtrace, I don't see my code in the stack.

If I comment out line 446 of behavior_path_planner_node.cpp

```
void BehaviorPathPlannerNode::run()
```

it will not crash but of course autoware will not enter autonomous state either.

Additional context

No response

felixf4xu commented on Jan 31 Author

from the same node `[component_container_mt-28]`, the last log before crash is:

```
[INFO] [1706696104.509989803]
[planning.scenario_planning.lane_driving.behavior_pl
register task: module = out_of_lane, id = 0
```

felixf4xu commented on Jan 31 Author

I have added some log into `behavior_path_planner_node.cpp` file, the output before crash is kind of random: maybe there are many threads running but the logs are from different threads so in the terminal, it's kind of random

maxime-clem added component:planning meeting:planning-control-wg labels on Jan 31

maxime-clem added this to **Planning & Control Working Group** on Jan 31

maxime-clem self-assigned this on Feb 1

maxime-clem commented on Feb 1 — Contributor

Thank you for reporting the issue and for the initial investigation.
It looks like you are using a custom map and I am not able to reproduce the issue on the sample map. Are you able to share your map ?
Otherwise, I can assist you in debugging the issue. First, I would recommend running the `behavior_planning_container` in a separate terminal (and ideally with `gdb`). This can be done by adding a `launch-prefix` in the behavior_planning.launch.xml like this:

```
<node_container pkg="rclcpp_components" exec
```

You should also make sure you build the `behavior_path_planner` with debug symbols:

```
colcon build --cmake-args -
DCMAKE_BUILD_TYPE=RelWithDebInfo --packages-
select behavior_path_planner
```

If you now reproduce the issue, you should be able to use `gdb` in the separate terminal to investigate the crash in more details.

❤️ 1

felixf4xu commented on Feb 2 — Author

Thanks for the comment, the crash is not related to the map, I uploaded a screenshot of the crash on the original map from autoware installation.
Screencast from 02-02-2024 06:50:49 PM.webm
In the screenshot, I modified the code just a little bit: `BehaviorPathPlannerNode::run()` is a timer callback, originally it is called every **100ms**, which makes it difficult to debug. So I changed the interval to **20 seconds**. So in the screenshot, you can see the delay.

- 00:09: I set the pose
- 00:14: since the timer callback is set to 20 seconds, so the function of `BehaviorPathPlannerNode::run()` is not called right after the setting of the goal but delayed

to be called at the time. The 'Auto' button is enabled. (but it's disabled again after a very shot while, which is also strange to me)

- 00:52 I waited for another cycle (20 seconds), then I re-set the goal
- 01:13 `BehaviorPathPlannerNode::run()` is called again, and the crash log is shown in the terminal.

maxime-clem commented on Feb 2 Contributor

I cannot reproduce the issue and I can think of 2 possible reasons:

1. you are using a version of Autoware with a bug, try updating your branches (`vcs pull src` in your autoware workspace).
2. The bug may only be reproducible with Eprosima DDS.

Please check 1 and I will check 2.

> The 'Auto' button is enabled. (but it's disabled again after a very shot while, which is also strange to me)

There are safeguards to disable the autonomous mode if a module takes too much time to publish its output. With a delay of 20s it is expected that the autonomous mode will be disabled.

maxime-clem commented on Feb 3 Contributor

I have been able to reproduce the issue with Eprosima DDS.

```
sudo apt install ros-humble-rmw-fastrtps-cpp
RMW_IMPLEMENTATION=rmw_fastrtps_cpp ros2 launc..
autoware_launch planning_simulator.launch.xml
map_path:=$MAP_PATH vehicle_model:=$VEHICLE_MODEL
sensor_model:=$SENSOR_MODEL
```

The issue does not seem to occur when launching the `behavior_path_planner` as a normal `node` (instead of the current `composable_node`). I do not understand the problem but I guess there is some problem with memory access when `behavior_path_planner` receives a new route.

More investigation will be required and in the meantime I recommend using another DDS if you can.

felixf4xu commented on Feb 3 Author

> launching the `behavior_path_planner` as a normal `node` (instead of the current `composable_node`)

Can you share how should I do this? I'm considering the same solution but don't know how to change the startup scripts/configs.

maxime-clem commented on Feb 3 — Contributor

> Can you share how should I do this?

Here is a commit with the change: maxime-clem/autoware.universe@ `8856575`

felixf4xu commented on Feb 4 • edited — Author

> I have been able to reproduce the issue with Eprosima DDS

there seems to be a similar issue at autowarefoundation/autoware.universe#5221 (comment), I'm not very sure if DDS is the root cause.

btw, can some one move this issue to `https://github.com/autowarefoundation/autoware.universe/issues`, I just realized that it should be there

idorobotics moved this to **In Progress** in **Planning & Control Working Group** on Feb 6

maxime-clem mentioned this issue on Feb 19

behavior_planning_container dies because of guard_condition — Closed

autowarefoundation/autoware.universe #6452

3 tasks

felixf4xu closed this as completed on Feb 29

github-project-automation bot moved this from **In Progress** to **Done** in **Planning & Control Working Group** on Feb 29

maxime-clem commented on Feb 29 — Contributor

@felixf4xu I do not think this issue is solved. Did you close it to move it to the `universe` issues ?

felixf4xu commented on Mar 1 — Author

@maxime-clem yes I know it's not solved, but I didn't see any more actions taken, so I think it's better to close it.

I also see you linked this issue to `https://github.com/autowarefoundation/autoware.universe/issues/6452` then I think it's safe to close this one.

For anyone else interested, my current workaround is using `RMW_IMPLEMENTATION=rmw_cyclonedds_cpp`.

luojiaxiang11 commented on Jun 19

> @maxime-clem yes I know it's not solved, but I didn't see any more actions taken, so I think it's better to close it.
>
> I also see you linked this issue to `https://github.com/autowarefoundation/autoware.universe/issues/6452` then I think it's safe to close this one.
>
> For anyone else interested, my current workaround is using `RMW_IMPLEMENTATION=rmw_cyclonedds_cpp`.

Hello have you solved this problem? I have encountered this problem as well, but it only occurs occasionally without a specific trigger scenario. I have setted RMW_IMPLEMENTATION=rmw_cyclonedds_cpp

felixf4xu commented on Jul 29 • edited — Author

> Here is a commit with the change: maxime-clem/autoware.universe@8856575

Can we merge this commit? I have several dev environment (different PC hardware, Ubuntu version, docker, none-docker) all have the crash issue and the workaround in the commit is the only way to fix it.

felixf4xu reopened this on Jul 29

maxime-clem commented on Jul 30 — Contributor

The node container is used for performance reasons so I do not think we want to merge that commit sorry.
I will try to find another solution.

maxime-clem commented on Aug 2 • edited ▾ Contributor

@felixf4xu the issue seemed to be coming from `rclcpp` and `rmw_fastrtps_cpp`.
This looks like the same issue: ros2/rmw_fastrtps#728
A fix PR exists for the `rolling` branch of `rclcpp` but I am not sure if it is on other branches as well 🤔
ros2/rclcpp#2142

maxime-clem commented on Aug 2 Contributor

@felixf4xu I tested with a locally build `rclcpp` package and could not reproduce the issue.
I was using this version:
https://github.com/tier4/rclcpp/tree/t4-main
So the fix seems to be on the `rclcpp` side.

felixf4xu commented on Aug 2 Author

Great, thanks!

felixf4xu closed this as completed on Aug 2