# Investigation on using Large Scale Lanelet2 Map in Autoware | Planning Simulator #4075

**StepTurtle** started this conversation in **General**

**Category**

💬 General

**Labels**

type:question
component:map
component:planning

**5 participants**

**StepTurtle** on Dec 20, 2023   Collaborator

Hello,

We are trying to use Autoware in large scale areas. For this purpose, we are working on how can we create large scale lanelet2 maps and can we use large scale lanelet2 maps with Autoware planning. In this discussion I want to share planning simulator results under large scale maps and I want to ask couple of questions.

First of all when we use large scale lanelet2 in current planning pipeline, there is an issue. When we load the lanelet2 map and select a route, if the route contains a lot of lanelet2 node, planning cannot work as expected, I'm mean cannot work with 10 hz. Also the FailSafe threshold is 5 hz for planning. When the route is approximately 400 kilometers, planning works with ~4 hz and the system does not allow autonomous driving. The following image shows the Runtime Monitor output for this situation.
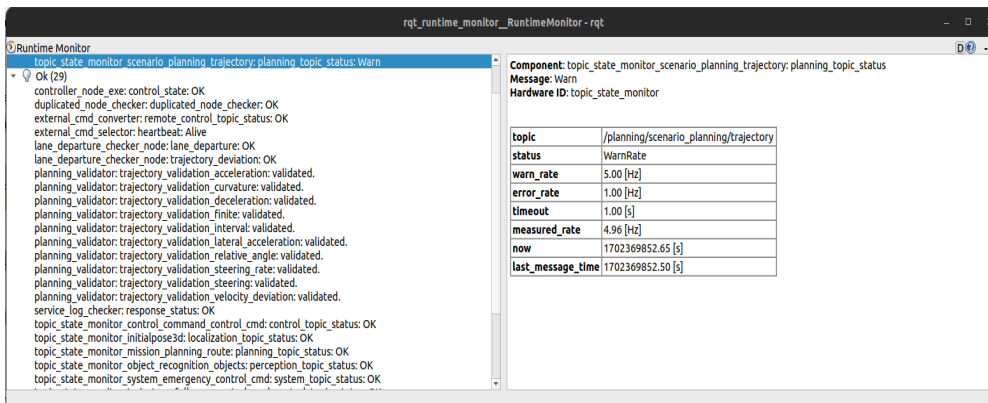


**The following test videos shows the frequency problem in two different system:**

- For this test we use dummy lanelet2 maps to avoid problems in lanelet if it exist. We used [this tool](#) to create dummy lanelet2 maps (it is developed from [kminoda](#) from tier4).
- The lanelet2 size is 14.0 mb and it contains 400 km road.

| ID | Video Link | CPU | Memory | GPU |
|---|---|---|---|---|
| Test 1 | https://youtu.be/yrFhXr40rXY | 12th Gen i7-12700H | 16 GB | RTX 3050 |
| Test 2 | https://youtu.be/0-z-yjgJaNA | 5900X (12c24t) | 64 GB | RTX 4080 |

- It can be see that, if we have bigger route in planning, the execution time of the code increases and after a while it drops below 10 Hz. The situation is valid in two different systems.

**To see the parts of the code that consume a lot of time, we debug the code and you can see the results in the schema:**

- You can use this link to see original schema. You can also click the link in the upper right corner to see the code in Autoware.



**When we examine the code, we can see there are two main part which waste a lot of time when we have a large route.**

1. First one is this function call. If I understand correct we are using `std::numeric_limits<double>::max()` as a `forward_length` parameter instead of this parameter to get all forward lanelet but it wasting a lot of time. I also try `300`, `1000`, `3000` etc. and this function used `0 ms` instead of `~100 ms`. I did not see any difference in my test but I am not sure is it okay or not?

2. Second one is this while loop. It is also iterating all lanelet and it is wasting a lot of time. I cannot find any solution to reduce time consumption but when I comment out this function call, planning simulator works same. Is this something that definitely needs to be worked, or is there a chance that we can close it?

If you have any idea on this topic, I will be very happy to hear.

↑ 2   👍 4   ❤️ 1   👀 1

---

**4 comments · 5 replies**

Oldest | Newest | Top

**maxime-clem** on Dec 21, 2023 `Collaborator`

@zulfaqar-azmi-t4 will look into the turn signal performances.

↑ 1  👍 1  👀 1                                                    1 reply

**maxime-clem** on Apr 2 `Collaborator`

I opened a PR to fix the turn signal performance: autowarefoundation/autoware.universe#6725

👍 1  👀 1

---

**StepTurtle** on Dec 22, 2023 `Collaborator` `Author`

PR related with left part (green part) of schema:

- ⚎ fix(behavior_path_planner): change getLaneletSequence function parameter for large scale lanelet2 autoware.universe#5940

↑ 2                                                                1 reply

**maxime-clem** on Apr 10 `Collaborator`

I reused your idea but changed the logic of the `PlannerManager` to use a moving `current_route_lanelet` for the base lanelet in the `getLaneletSequence` call, fixing the issue of the previous PR: autowarefoundation/autoware.universe#6739

👍 1

---

**yukkysaito** on Dec 24, 2023 `Maintainer`

Thank you for the information. Regarding TIER IV, We are still in the process of validating maps of such large scale. There are two major issues identified:

1. Due to the precision limitations of the float data type, when dealing with several hundred kilometers, the accuracy to the order of a few centimeters becomes a problem. It's understood that several codes currently represent positions using the float data type.
   Autoware (especially the Lanelet2 library) does not anticipate maps of such large sizes. Consequently, the computational costs of some search functions increase significantly.
2. Regarding the second issue, TIER IV's Planning team is currently investigating it (cc @takayuki5168 @kminoda ). In a few months, they expect to handle maps of about 100 kilometers without any problems, as long as the maps have few objects.

However, for maps exceeding several hundred kilometers, significant modifications will be necessary. Depending on the situation, a new map format or library might be required. Currently, discussions are ongoing with @xmfcx and @mitsudome-r .

↑ 3      👍 1      👀 1                                    3 replies

**xmfcx**  on Dec 26, 2023    Maintainer

Hello @yukkysaito -san

Thank you for sharing the challenges faced with large-scale maps in Autoware and the use of the Lanelet2 library. Your concerns about the precision limitations and computational costs are valid in the context of large-scale routing. Let me offer some insights and potential solutions:

## Potential Solutions and Resources:

1. **Fast Route Planning Techniques:**

   - **Google TechTalks (March 23, 2009):** [Fast Route Planning](#) offers insights into efficient algorithms for route planning.
   - **Engineering Route Planning Algorithm:** The [accompanying slides](#) provide a deeper understanding of the algorithms discussed.
   - **Transit Node Routing:** As mentioned on [Wikipedia](#), this approach can significantly reduce query times for route planning.

2. **OpenStreetMap (OSM) for Large-Scale Planning:**

   - OSM demonstrates the feasibility of large-scale route planning. For example, [OSM Directions](#) can quickly recalculate routes even over long distances. (Drag the pins around to see how quickly the route is recalculated.)
   - **Open Source Services/Libraries:**
     - [OpenRouteService](#) and its [GitHub repository](#) show how large-scale routing can be implemented.

## Lanelet2 and OSM:

- **Lanelet2 & OSM Integration:** Lanelet2, being an extension of [OSM_XML](#), already provides a strong foundation for handling large-scale maps. The challenge lies in optimizing Lanelet2 to handle the computational complexity efficiently.
- **Scalability with OSM:** Given OSM's capability to support large-scale maps and routing, it should be plausible to adapt and scale Lanelet2 similarly. This would involve exploring more efficient data structures or algorithms to reduce the computational overhead.

cc. @TakaHoribe @takayuki5168 @mehmetdogru @mitsudome-r

**xmfcx**  on Dec 26, 2023    Maintainer

We should investigate https://github.com/GIScience/openrouteservice and see if it can replace the Mission Planner in any way.

**ataparlar**  on Dec 26, 2023   Collaborator

Hello **@yukkysaito** -san

Thank you for your thoughts in this topic. According to our observations, we can support the issues determined. However, we don't think that we need to change the libraries or map format. Mission planner's performance depends on the area size due to the floating points and it has an inverse proportion with node count in the Lanelet2 map. Instead of loading all the map for route planning for large areas, we should consider keeping the map area lower and change the map area in the Autoware environment. Basically, we can call that Dynamic Lanelet2 Map Loading.

The first thing that we thought is using some service like Open Route Service to plan the route for larger areas. After that we can consider that plan as a linestring that the road we will cover. We can take goal points in every 5 kilometers from this linestring for example. Still we need to change somethings in planning pipeline and we should consider to make it compatible with the areas that over 100km. We will publish a proposal for dynamic lanelet2 loading related to this issue and that issue: autowarefoundation/autoware.universe#5959.
Thank you for your attention.
cc: **@StepTurtle**

**xmfcx**  on Dec 26, 2023   Maintainer

Related Issues:

- ⊙ **Feature: Remove Autoware limitations about map boundaries** autoware.universe#5959

↑ 1                                                                 0 replies