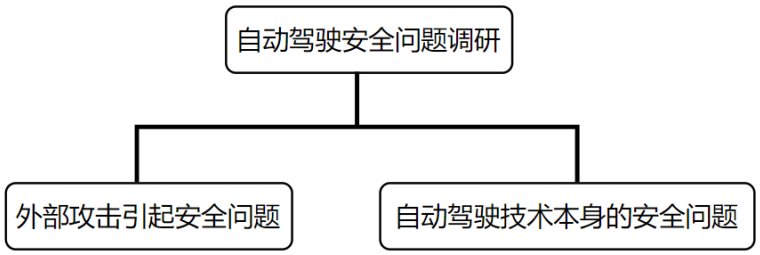
目录

自动驾驶安全问题调研

概要设计:



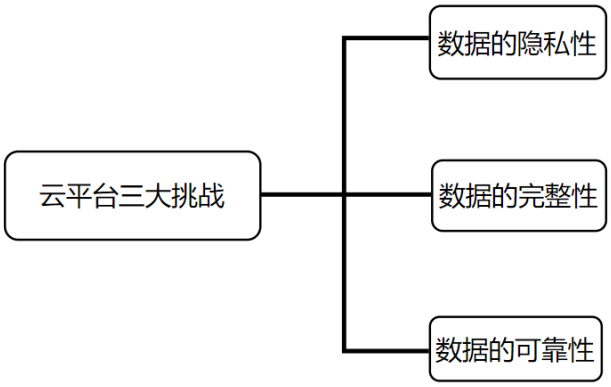
自动驾驶安全问题调研可大概分为 2 个问题：

- 外部攻击引起安全问题
- 自动驾驶技术本身的安全问题

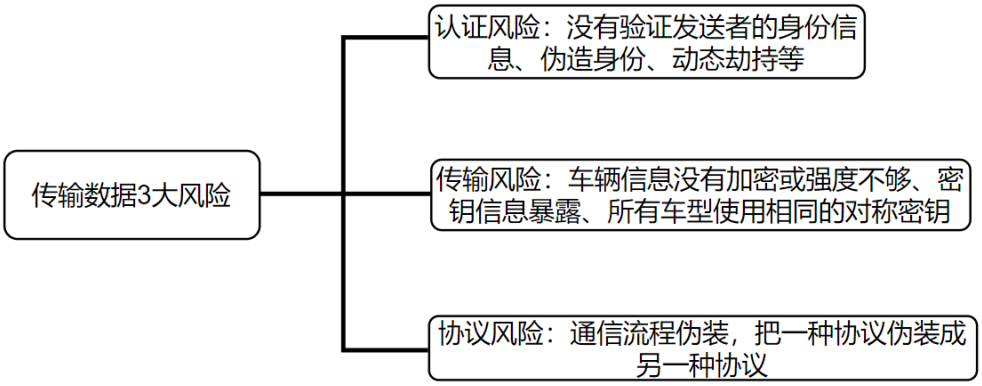
详细设计：

外部攻击安全问题

1. 云端威胁



2. 传输威胁

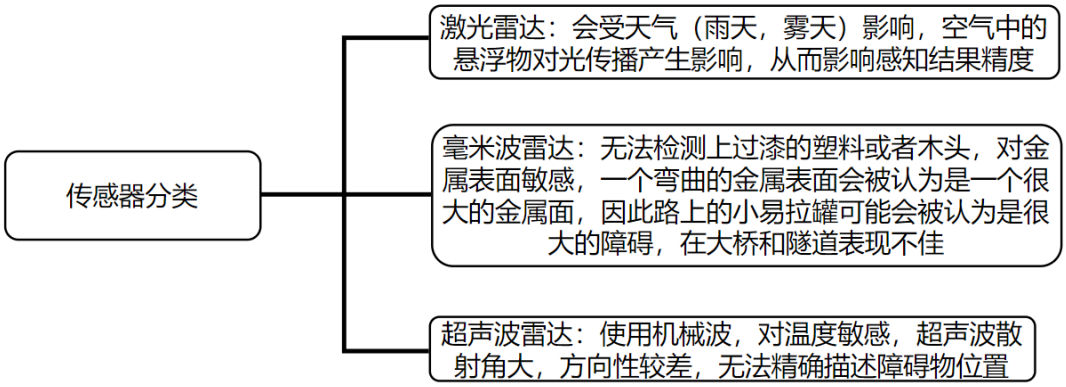


3. 终端威胁
   - T-BOX 的安全威胁：T-BOX 实现车内网和车际网之间的通信
   - IVI 安全威胁：IVI 的高集成度使其所有接口都有可能成为黑客的攻击节点
   - 终端升级安全威胁：OTA 升级过程中也面临升级包被劫持风险
   - 车载 OS 安全威胁：操作系统本身可能有安全漏洞
   - 接入风险：通过 OBD 接口监听总线上面的消息并且伪造消息来欺骗 ECU
   - 无线传感器安全威胁：传感器存在通讯信息被窃听、中断、注入等威胁
4. 移动 APP 威胁

市场上大多数智能网联汽车远程控制 APP 安全强度不够，对那些 APP 进行逆向分析挖掘，可以发现 APP 内的核心内容，包括存放在 APP 中的密钥，重要控制接口等。

自动驾驶技术本身的安全问题

1. 感知系统的安全问题



2. 感知融合算法的安全问题
   - 需要多个传感器之间取长补短，来提升定位的正确率和精确度
   - 由于环境的复杂多变，传感器可得到的数据十分多样，某些情况下当前使用的感知融合方案可能会出错
3. 决策模块的安全问题

决策模块功能：在一套完整的自动驾驶系统中，如果将感知模块比作人的眼睛和耳朵，那么决策规划就是自动驾驶的大脑。

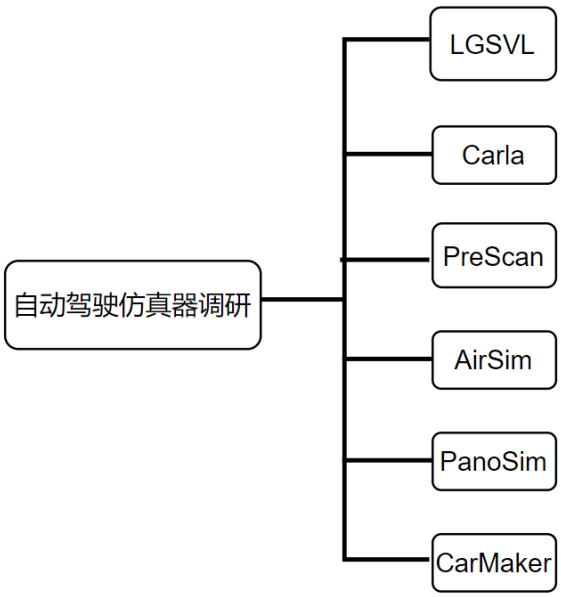
大脑在接收到传感器的各种感知信息之后，对当前环境作出分析，然后对底层控制模块下达指令，这一过程就是决策规划模块的主要任务。

安全问题：在应对环境多变性、检测不准确性、交通复杂性以及交规约束性等诸多车辆行驶不利因素时，决策模块未必能作出正确的决策。

综合车辆运行环境及车辆信息，结合行驶目的做出具有安全性、可靠性以及合理性的驾驶行为是决策的难点，亦是实现自动驾驶的难点。

自动驾驶仿真器调研

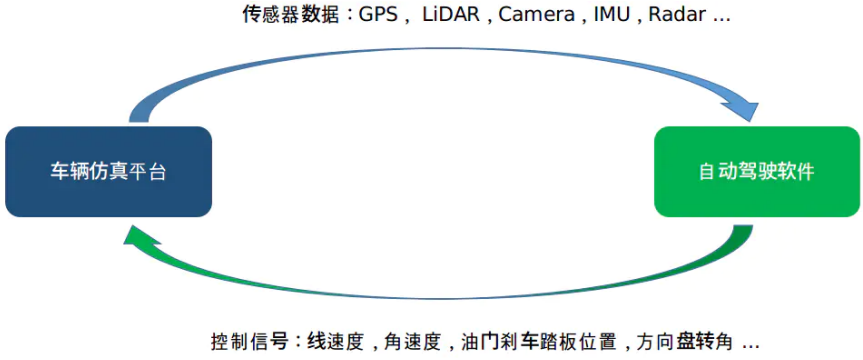
概要设计：



详细设计：

LGSVL
简介：
开源软件，免费使用。
基于游戏引擎 Unity 开发的一款主要用于自动驾驶开发和测试，支持包括仿真环境、传感器以及通讯内容的自定义。
LGSVL 由模拟软件、软件工具、支持定制用例的内容和插件生态系统以及支持大规模模拟和场景测试的云环境组成，允许开发人员进行调试、执行模块化测试和执行集成测试。
支持与其他自动驾驶软件进行闭环仿真测试：

-Apollo
-Autoware



Carla

简介：

Carla 模拟器由可扩展的客户端-服务器架构组成。

服务器负责与模拟本身相关的所有事情：传感器渲染、物理计算、世界状态及其参与者的更新等等。

客户端由一组客户端模块组成，这些模块控制场景中行为者的逻辑并设置世界条件。这是通过利用 CARLA API（在 Python 或 C++ 中）来实现的。

核心模块：

- Traffic manager：模拟类似现实世界负责的交通环境
- Sensors：模拟真实世界的传感器模型，包括相机、激光雷达、声波雷达、IMU、GNSS 等
- Recorder： 用来记录仿真每一个时刻的状态，可以用来回顾、复现
- ROS bridge and Autoware implementation：让 Carla 与 ROS 还有 Autoware 交互
- Open assets： 提供了不同的地图，可以控制天气条件，并提供经常使用的参与者模型
- Scenario runner：提供参考实例以便的搭建一个需要的测试场景

PreScan

简介：

PreScan 是西门子公司旗下汽车驾驶仿真软件产品，PreScan 是以物理模型为基础，开发 ADAS 和智能汽车系统的仿真平台。

支持摄像头、雷达、激光雷达、GPS，以及 V2V/V2I 车车通讯等多种应用功能的开发应用。

PreScan 基于 MATLAB 仿真平台，主要用于（ADAS）汽车高级驾驶辅助系统和无人自动驾驶系统的仿真模拟软件，其包括多种基于雷达，摄像头，激光雷达，GPS，V2V 和 V2I 车辆/车路通讯技术的智能驾驶应用。

使用方法：

- PreScan 会结合 Matlab/Simulink、Carsim 一起使用
- 用 Matlab/Simulink 做控制算法的开发
- 用 Carsim 仿真软件提供汽车动力学模型、轮胎模型和制动器模型
- 利用 Prescan 软件建立测试场景与传感器模型
- 其中 PreScan 软件和 MATLAB/Simulink 软件可以相互调用，具体来说就是 PreScan 中的各种传感器仿真数据传递到 Simulink 中

CarMaker

简介：

CarMaker 是德国 IPG 公司旗下的一款专注于乘用车的动力学仿真软件，它包括了精准的车辆本体模型(发动机、底盘、悬架、传动、转向等)，同时也可以提供包括车辆，驾驶员，道路，交通环境的闭环仿真系统。

CarMaker 既可针对开发流程前期进行的模型在环、软件在环等离线仿真，也可以接入 ECU、子系统总成、网络等做硬件在环测试。

包含道路三维模型，交通环境模型以及适应不同驾驶环境以及不同驾驶策略的驾驶员模型。

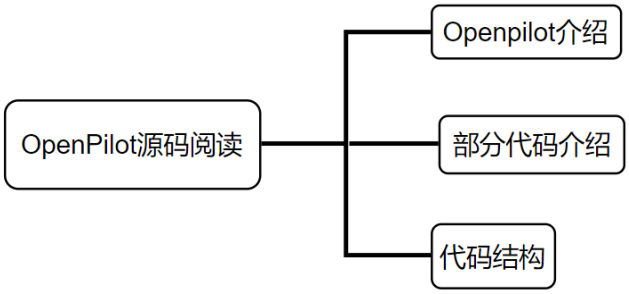
特性：

- 支持高精地图的导入与导出
- 支持在高性能计算 (HPC) 集群上并行执行大量测试目录
- 支持在 Docker 容器中运行
- 具有良好的可移植性和可扩展性等特点

- 可以与很多第三方软件进行集成， 如 ADAMS、AVLCruise、rFpro 等，可利用各软件的优势进行联合仿真。
- CarMaker 配套的硬件，提供了大量的板卡接口， 可以方便的与 ECU 或者传感器进行 HIL 测试。

OpenPilot 源码阅读

概要设计:



详细设计:

OpenPilot 介绍

- OpenPilot 是由 comma.ai 开发的开放源代码半自动驾驶系统。OpenPilot 可以代替 OEM 的高级辅助驾驶系统，用来改善视觉感知与机电执行器控制。它让用户可以透过增加的计算能力、强化的侦测器以及不断更新的驾驶辅助功能来修改现有的汽车，这些功能会随用户递交的资料而持续改善。

- 功能

（1）自动车道置中

OpenPilot 使用经过驾驶用户资料训练过的机器学习来决定道路上最安全的路径。这可以改善在没有车道标线的道路上行驶的表现，并透过追踪前方的车道线来维持车道置中。

（2）主动式车距维持定速

OpenPilot 能与前前方车辆保持安全的跟车距离。它可以在无用户干预的情况下以随停随走的方式行驶。它使用开放街图的道路曲率与速度限制资料以让车辆在急转弯时放慢速度，并将车辆的速度维持在目前的速限之下。

（3）驾驶监控

OpenPilot 会监控驾驶脸部，如果驾驶分心了则会被警示。如果驾驶分心超过六秒，OpenPilot 会将车辆减速至停止，并以声音对用户发出警报。

（4）车道变换辅助驾驶

开启方向灯时，OpenPilot 会使用此功能来变换车道，驾驶必须在方向盘上进一步动作以确认变换车道。在部分的品牌与车款上，OpenPilot 也会与盲点警示系统交互，以在盲点警示系统侦测到其他车辆时阻止变换车道。

OpenPilot 代码结构

```
├── apk 用于 UI的 apk文件
├── cereal 用于 EON上所有日志的消息传递规范
├── common 我们在这里开发的库函数
├── installer/updater 管理 openpilot的自动更新
├── opendbc 如何解读汽车数据的文件
├── panda 用于在 CAN和 LIN上通信的文件
├── phonelib EON上使用的库
├── pyextra EON上使用的库
├── third_party 外部引进的库
├── selfdrive 驾驶汽车的代码
```

```
├── Debug 帮助您调试和执行汽车端口的工具
├── Locationd 精确定位
├── Logcatd 从 andriod获取 logcat信息
├── Loggerd 汽车数据记录器和上传器
├── Proclogd 进程记录信息
├── Sensord IMU/GPS接口代码
├── Test 汽车模拟器通过虚拟操作运行代码
├── Ui The UI
├── Visiond 视觉通道-与相机交谈，运行模型，保存视频
├── Manager.py 管理进程
├── Messaging.py 消息传递
├── Thermal.py CPU,GPU等散热
```

部分代码介绍

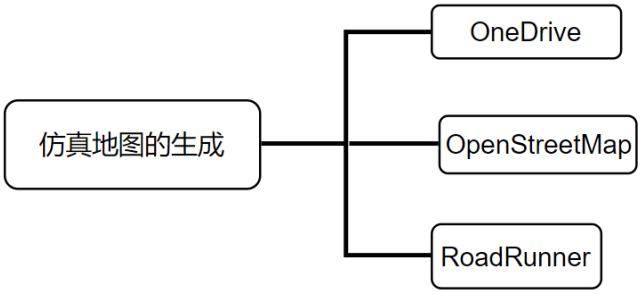
selfdrive\lib 中包括了控制车辆进行自动驾驶辅助工作的代码，可以帮助车辆进行横向纵向控制，进行路径规划，管理预警系统。

```
├── lateral_mpc横向的汽车 ADAS摄像头
├── longitudinal_mpc纵向的汽车 ADAS摄像头
├── alertmanger.py 报警管理-程序
├── driver_helpers.py 驾驶助手
├── latcontrl.py 横向控制-程序
├── latcontrol_helpers.py横向控制-驾驶助手
├── longcontrol.py 纵向控制-油门和刹车
├── pathplanner.py路径规划器-决定在哪里开车
├── pid.py PID算法库
├── planner.py 规划器
├── radar_helpers.py 雷达助手
├── speed_smother 速度平滑库
├── vehicle_model.py 车辆型号
├── controlsd.py 实际上驾驶汽车
├── radard.py 处理雷达数据
```

- longitudinal_planner.py
- 用于纵向规划，首先根据插值得到允许的最大加速度，再由横向加速度和总加速度得到纵向加速度。根据当前车速计算出下一次迭代时的速度，并不得超过期望速度。当在纵向方向加完速后，如果有需要，强制进行一个平滑的降速。
- lateral_planner.py
- 它用以更新驾驶车在横向方向的状态。首先检查当前车辆的车道变换状态，如果车道变换状态为开启且车道变换不超时，则根据车辆左右信号灯情况选择做右车道进行变换。如果驾驶车的车道变换状态为关闭或与上次车道变换状态相同，则意味着将进行新的车道变换周期，于是将车道变换的计时归为 0.
- planner.py
- 在 plannerd.py 文件中，分别调用了横向方向的规划和纵向方向的规划。它决定了汽车接下来的形式位置。首先日志中获取汽车参数，并根据参数决定是否使用车道线以及广角相机，接下来不断地调用 longitudinal_planner 和 laternal_planner 来对纵向和横向方向的移动进行规划，进而组合成现实世界中的自动驾驶规划。
- controlsd.py
- 是对汽车进行总体控制的主要代码，它执行一个 100hz 的主要循环，采用发布-订阅消息的模式接受并发送与汽车运行相关的数据以及信息。根据订阅的消息对汽车进行横向，纵向的控制，规划路径等，并把控制消息返回给其他程序，使得汽车得以安全可靠地运行。
- 发布的消息包括 controlState，carState，carControl，carEvents，carParams 等，订阅的消息包括 deviceState, longitudinalPlan, lateralPlan, deviceState, managerState, pandaState 等。

仿真地图的生成

概要设计:



作为一个完整的仿真测试场景描述方案，OpenX 系列标准包括：OpenDRIVE、OpenSCENARIO 和 OpenCRG。仿真测试场景的静态部分（如道路拓扑结构、交通标志标线等）由 OpenDRIVE 文件描述，道路的表面细节（如坑洼、卵石路等）由 OpenCRG 文件描述，仿真测试场景的动态部分（如交通车的行为）由 OpenSCENARIO 文件描述。

大部分仿真器都是支持 OpenDRIVE 的。支持 OpenSCENARIO 的软件，目前看到的有 VTD 和 Carla、51Sim-One(51Sim-One1.2, 2020 年发布)、Prescan(PreScan2019.3 版本开始)。LGSVL 暂时没有资料显示支持 OpenSCENARIO。

仿真器	是否支持 OpenDrive	是否支持 OpenSCENARIO	版本
VTD	支持	支持	VTD 2.2
Carla	支持	支持	Carla 0.9.8
51Sim-One	支持	支持	51Sim-One 1.2
preScan	支持	支持	preScan 2019.3
Airsim	没有资料显示支持	没有资料显示支持	——
LGSVL	支持	没有资料显示支持	LGSVL 2020.05
AADS	没有资料显示支持	没有资料显示支持	——
TAD Sim	没有资料显示支持	没有资料显示支持	——

详细设计:

OpenDRIVE

- OpenDRIVE 是对路网结构的描述性文件，OpenDRIVE 将道路（roads）分为三个部分：道路参考线（reference line）、车道（lanes）和道路设施（features）。除此以外，还可以设置道路的高度 (elevation)，对于多条道路汇聚的位置需要用路口 (junctions) 来描述。
- 官方文档中提供了一个 OpenDRIVE 独立模式，Carla 允许用户将任何 OpenDRIVE 文件作为现成的 Carla 地图摄取。为了做到这一点，模拟器将自动生成一个道路网格供 actor 导航。此模式仅使用 OpenDRIVE 文件运行完整模拟，而不需要任何其他几何图形或资源。为此，模拟器获取一个 OpenDRIVE 文件，并按程序创建一个临时三维网格来运行模拟。生成的网格以极简的方式描述道路定义。所有元素都将与 OpenDRIVE 文件相对应。交通信号灯、停车场和停车场将实时生成。行人将在地图上显示的人行道和人行横道上运动。所有这些元素，以及路上的每一个细节，都是基于 OpenDRIVE 文件的。

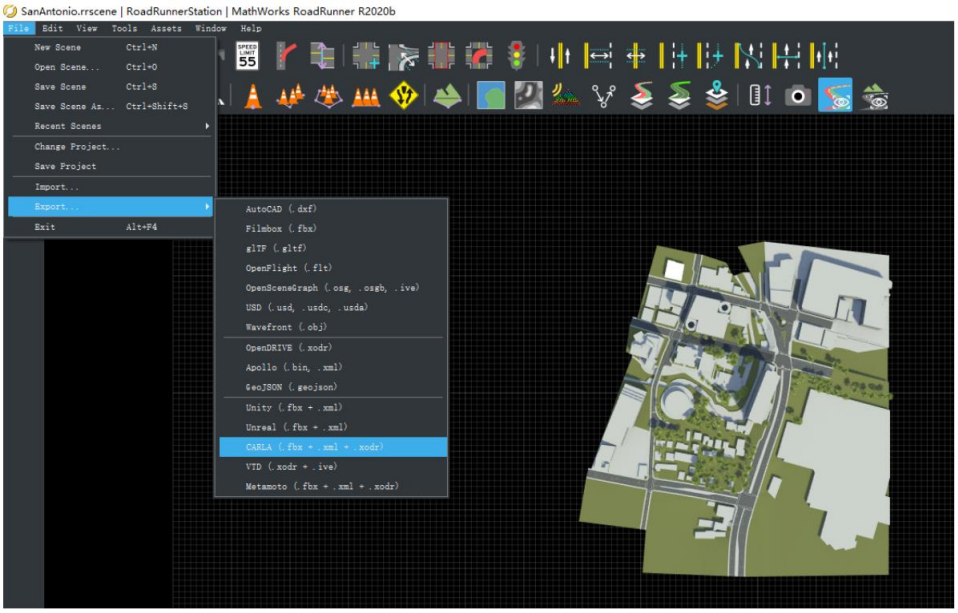
OpenStreetMap

- OpenStreetMap 是开源的世界开放许可证地图。这些地图的各个部分可以导出为 XML 格式的 .osm 文件。CARLA 可以将文件转换为 OpenDRIVE 格式，并使用 OpenDRIVE 独立模式导入它。
- 整个过程如下:

- 使用 OpenStreetMap 获取地图
- 转换为 OpenDRIVE 格式
- 导入 Carla

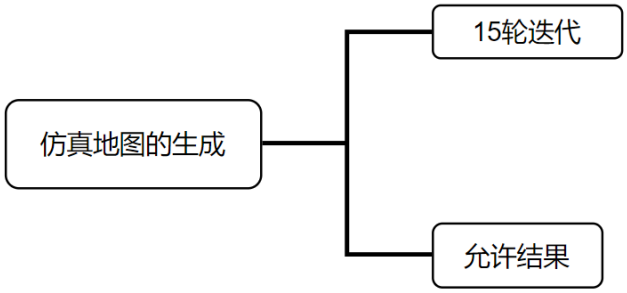
RoadRunner

- RoadRunner 使任何人都可以轻松创建环境，逼真的道路和道具。
- 可以用于生成包含回旋处，交叉路口和桥梁的复杂道路网络，自定义标牌，标记和道具，然后发布到 OpenDRIVE，Unreal，Unity 或 FBX 等，以实现全面的灵活性。
- 现在可用于 Windows（7x64 或更高版本）Linux（Ubuntu x64 16.04）以及更多版本



Carla 中模拟撞击事故

概要设计:



详细设计:

- 通过在路上画黑色线条的方法，从而伪造车道线，对 Carla 提供的经过训练的端到端模型进行攻击，使车辆偏离原方向。
- 使用 Carla 模拟器进行仿真。在实验的时候，只使用 RGB 相机，禁用激光雷达(光探测和测距)、语义分割和深度相机。通过虚幻引擎 4 (UE4)，创建了一个新的对抗性平面蓝图，这是一个 200×200 像素平面或画布与动态 UE4 材料，可以覆盖在所需的道路部分。这个蓝图的关键属性是读取生成的攻击图像(.png 文件)，并将其实时放置在 CARLA 中。
- 右转的风险最大，这意味着这样的场景最容易攻击
- 出现错误的原因之一是部分错误地将道路上的绘制线误认为是左转场景中常见的路缘或障碍物，从而导致车辆在本应右转的情况下急剧向左转向。

15 轮迭代

运行结果分析

```
nning the baseline scenario.
mplete.
nning the Bayesian Optimizer for 15 iterations.
  iter    |  target   |   pos1    |   pos2    |   rot1    |   rot2    |
-------------------------------------------------------------------------
ROR:root:(127.0.0.1:2000) failed to read data: [Errno 104] Connection reset by peer
 1        |  6.234    |  71.16    |  190.1    |  131.0    |  107.2    |
ROR:root:(127.0.0.1:2000) failed to read data: [Errno 104] Connection reset by peer
 2        |  4.033    |  29.64    |  31.2     |  10.4     |  155.0    |
ROR:root:(127.0.0.1:2000) failed to read data: [Errno 104] Connection reset by peer
 3        |  2.974    |  114.2    |  141.6    |  3.685    |  173.6    |
ROR:root:(127.0.0.1:2000) failed to read data: [Errno 104] Connection reset by peer
 4        |  5.332    |  158.2    |  42.47    |  32.55    |  32.83    |
ROR:root:(127.0.0.1:2000) failed to read data: [Errno 104] Connection reset by peer
 5        |  14.75    |  57.81    |  105.0    |  77.32    |  52.13    |
ROR:root:(127.0.0.1:2000) failed to read data: [Errno 104] Connection reset by peer
 6        |  10.61    |  52.81    |  98.72    |  78.16    |  48.16    |
ROR:root:(127.0.0.1:2000) failed to read data: [Errno 104] Connection reset by peer
 7        |  12.06    |  50.8     |  84.63    |  78.8     |  48.49    |
 8        |  12.28    |  56.68    |  91.68    |  72.14    |  46.73    |
 9        |  12.15    |  64.67    |  97.37    |  70.78    |  51.6     |
ROR:root:(127.0.0.1:2000) failed to read data: [Errno 104] Connection reset by peer
 10       |  13.8     |  43.17    |  82.9     |  78.99    |  42.0     |
 11       |  14.55    |  60.71    |  110.7    |  76.86    |  55.15    |
 12       |  12.53    |  46.34    |  76.31    |  70.22    |  38.27    |
 13       |  11.77    |  63.11    |  106.7    |  83.01    |  52.79    |
 14       |  13.93    |  55.71    |  112.8    |  73.17    |  47.45    |
ROR:root:(127.0.0.1:2000) failed to read data: [Errno 104] Connection reset by peer
 15       |  14.59    |  37.94    |  78.45    |  71.7     |  42.39    |
=========================================================================
```