DORA-RS & DORA-DRIVES

OS2EDU

Dora-rs

Quick introduction

In 2023, Open Source AI is booming! Open Source Robotic however hasn't changed much in years...

From my perspective as a generalist software engineer who isn't a low-level robotics expert – as an ecosystem, **[ros2] seems to have adopted every bad practice available and invented some more of its own**. [...]. **Many of the architectural design decisions are frankly baffling**, although I appreciate that this is in part down to age, legacy, and the open nature of the platform.

- HackerNews Top comment

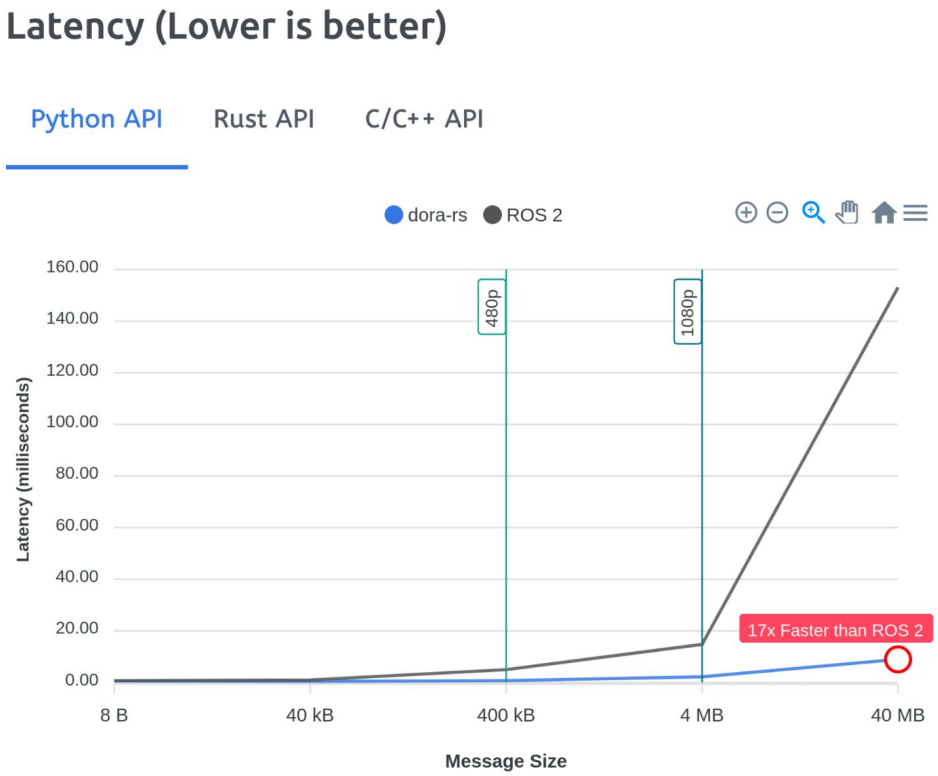
dora-rs

dora goal is to make DIY Robotics fast and simple

- Make AI in robotics simple to use
- Limit the use of external tooling and use Industry Standard General Purpose tooling
- Simple integration of Sensors in robotics.

https://github.com/dora-rs/dora

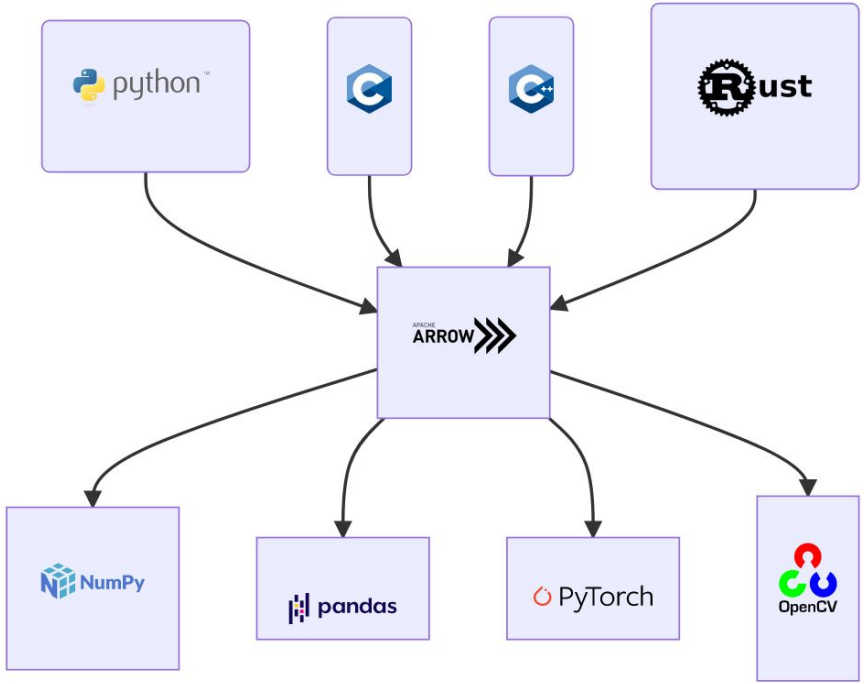
- How fast?


Latency (Lower is better)
Python API
Rust API
C/C++ API
dora-rs
ROS 2
160.00
140.00
120.00
100.00
80.00
60.00
40.00
20.00
0.00
Latency (milliseconds)
480p
1080p
17x Faster than ROS 2
8 B
40 kB
400 kB
4 MB
40 MB
Message Size

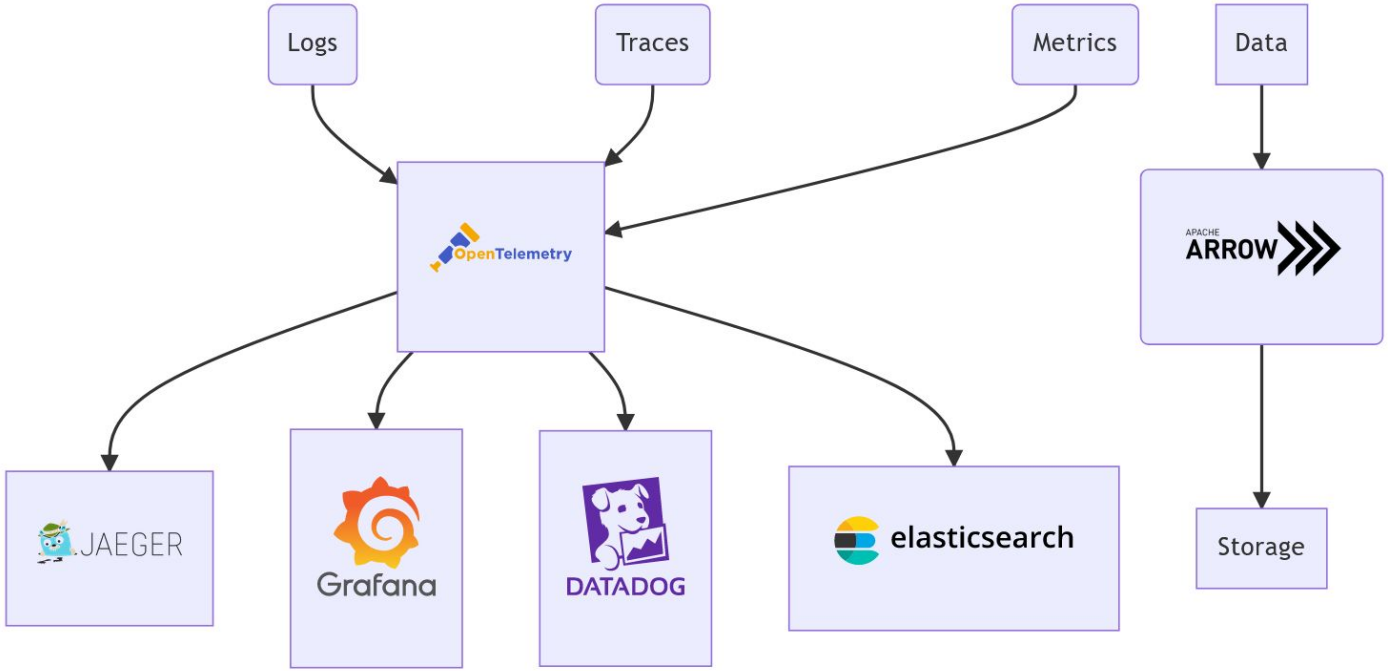
Own shared memory Server

- Written in **Rust**
- Implement **Zero-Copy**
- Easy to use on **many OS** and **programming language**
- **Finetuned to dora-rs**

Apache Arrow Format



OPENTELEMETRY



INDUSTRY STANDARD DEV ENV: HOT-RELOADING

- See change in real time
- while keeping state
- Similar to React, FastAPI

https://www.youtube.com/watch?v=lTzPTxCUFSM

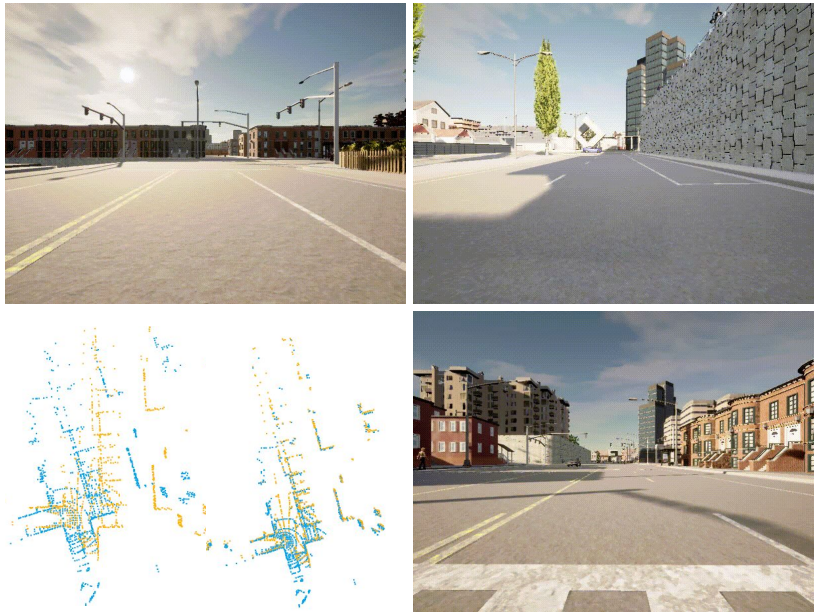
DISTRIBUTED CLUSTER

- Support dataflow on remote machine
- Future features:
  - Support fast prototyping on remote machine
  - Enable fast file-syncing between machine

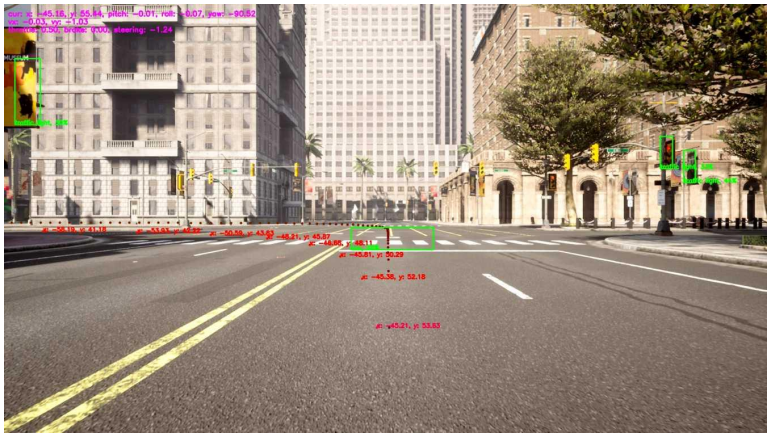
dora-drives

- Autonomous driving starter kit
  - Provides many driving building blocks:
    - Yolov5 Object detection
    - Yolop Lane and Drivable area detection
    - Strong Sort Object tracking
    - Frenet Optimal Trajectory Planning
    - IMFnet DL localization
    - …

https://github.com/dora-rs/dora-drives



Use Case #1: Carla Simulator



Carla Simulator

- Run **locally** and on **Server**
- Simulate **Sensors** and **Control**
- Simple SDK
- Testbed for DL Models and Algorithm

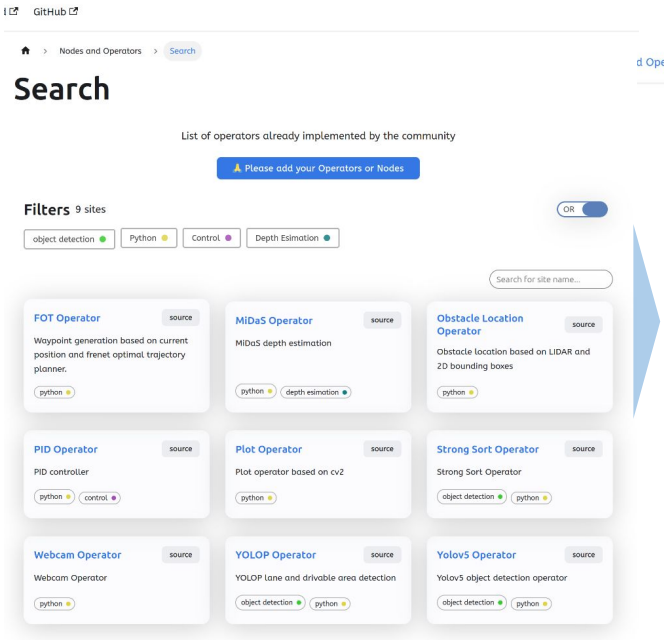
Use Case #2: Dora Rover

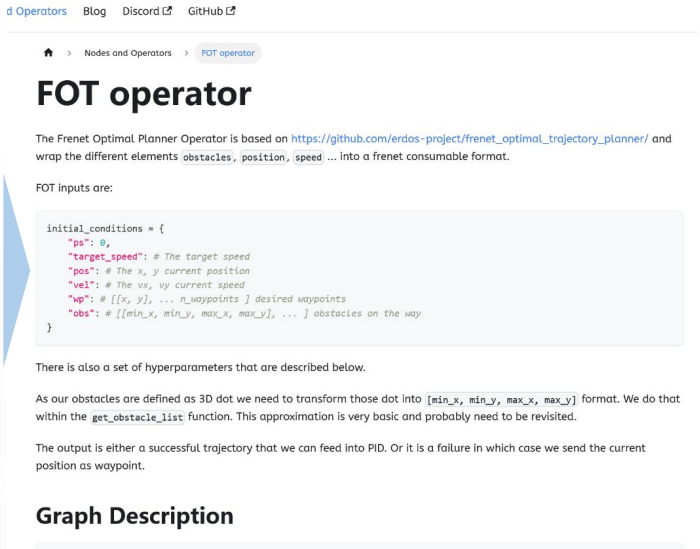
- Use Nvidia Orin
- Velodyne Lidar
- IMU, GNSS
- Camera
- Mavros controller
- Lidar-based localization

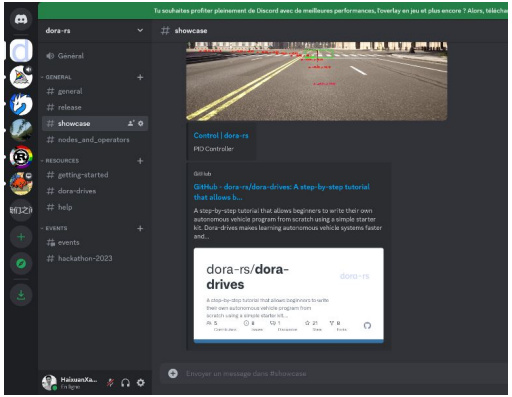
Use Case #3: Dora Car

- RockChip RK3588
- Chinese LIDAR
- 6 Cameras
- **Custom Drive-by-wire CANBUS**

Node HUB


GitHub
Nodes and Operators
Search
Search
List of operators already implemented by the community
Please add your Operators or Nodes
Filters 9 sites
OR
object detection
Python
Control
Depth Esimation
Search for site name...
FOT Operator
source
Waypoint generation based on current position and frenet optimal trajectory planner.
python
MiDaS Operator
source
MiDaS depth estimation
python
depth esimation
Obstacle Location Operator
source
Obstacle location based on LIDAR and 2D bounding boxes
python
PID Operator
source
PID controller
python
control
Plot Operator
source
Plot operator based on cv2
python
Strong Sort Operator
source
Strong Sort Operator
object detection
python
Webcam Operator
source
Webcam Operator
python
YOLOP Operator
source
YOLOP Lane and drivable area detection
object detection
python
Yolov5 Operator
source
Yolov5 object detection operator
object detection
python


d Operators
Blog
Discord
GitHub
Nodes and Operators
FOT operator
FOT operator
The Frenet Optimal Planner Operator is based on https://github.com/erdos-project/frenet_optimal_trajectory_planner/ and wrap the different elements obstacles, position, speed ... into a frenet consumable format.
FOT inputs are:
initial_conditions = {
"ps": 0,
"target_speed": # The target speed
"pos": # The x, y current position
"vel": # The vx, vy current speed
"wp": # [[x, y], ... n_waypoints] desired waypoints
"obs": # [[min_x, min_y, max_x, max_y], ...] obstacles on the way
}
There is also a set of hyperparameters that are described below.
As our obstacles are defined as 3D dot we need to transform those dot into [min_x, min_y, max_x, max_y] format. We do that within the get_obstacle_list function. This approximation is very basic and probably need to be revisited.
The output is either a successful trajectory that we can feed into PID. Or it is a failure in which case we send the current position as waypoint.
Graph Description


dora-rs
showcase
General
GENERAL
general
release
showcase
nodes_and_operators
RESOURCES
getting-started
dora-drives
help
EVENTS
events
hackathon-2023
Control | dora-rs
PID Controller
GitHub - dora-rs/dora-drives: A step-by-step tutorial that allows b...
dora-rs/dora-drives

PARTNER UNIVERSITY

CentraleSupélec
université PARIS-SACLAY
National Taiwan University
UNIVERSITY OF CAMBRIDGE

BACKWARD COMPATIBILITY

- dora-ros2-bridge
  - Auto-generate Custom Messages interface
  - Can run ROS2 node without compiling
  - Can use legacy Tooling such as rosbag, rviz, ...

ROADMAP

dora-rs 0.2
Rust, Python, C/C++
API
Arrow
Opentelemetry
ros2-bridge

→

dora-rs 0.3
Data Log & Replay
Data Visualization
Time Constraints
Deadline
Fault tolerance
Cloud Native Integration

→

dora-rs 0.4
Fault tolerance
Redundancy
Elastic
Resources
Dynamic
Dataflow

→

dora-rs 0.5
Fleet Management
Fleet Support
Mission Scheduling

Open for feature proposals

QR CODE

Github

Discord

Dora-rs

How to get started

https://dora.carsmos.ai/docs/guides/

THANK YOU