dora-rs

Modern Dataflow Framework for Robotics

Haixuan Xavier Tao, Founder of 1ms.ai
Motivation

• Make creation of robotic applications fast and simple
• Super easy integration with latest technologies (e.g. DL models, Cloud, DBs ...)

Use Case

• DIY off-the-shelf robotics hardware
• Research

Audience

• Robotics Enthusiasts
• Masters Students
• Phds
What is a robotic application?

Programming a robot can be summed up as bringing together hardwares, algorithms, and AI models, each with their main loop and make them communicate with each others.
Design

- Application are defined as dataflow graph
  - Nodes are separate process → isolation, flexibility
  - Each node defines a set of inputs and outputs

```python
defines:
- id: webcam
  path: webcam.py
  outputs:
    - image
- id: object_detection
  path: yolov8.py
  inputs:
    image: webcam/image
  outputs:
    - bounding_box
- id: plot
  path: dora-rerun
  inputs:
    bounding_box: object_detection/bounding_box
    image: webcam/image
```
Communication

Arrow based communication using zero copy and shared memory already integrated in your favorite libraries.
Performance

Over 20x improvement compared to existing open source solutions in Python

Latency (Lower is better)

- Python API
- Rust API
- C/C++ API

- dora-rs
- ROS 2

- 17x Faster than ROS 2

Message Size

8 B 40 kB 400 kB 4 MB 40 MB
## Deep Learning Models

- Simple integration with AI models:

<table>
<thead>
<tr>
<th>Model</th>
<th>Functionality</th>
<th>Latency</th>
<th>GPU RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>openai/whisper</td>
<td>Speech to Text</td>
<td>1.5s</td>
<td>0.7GB</td>
</tr>
<tr>
<td>huggingface/idefics2-8B</td>
<td>Image to Text</td>
<td>1.5s</td>
<td>8GB</td>
</tr>
<tr>
<td>huggingface/parler-600M</td>
<td>Text to Speech</td>
<td>1.5s</td>
<td>3GB</td>
</tr>
<tr>
<td>ultralytics/yolov8n</td>
<td>Image recognition</td>
<td>2ms</td>
<td>0.7GB</td>
</tr>
<tr>
<td>deepseek-ai/DeepSeek-Code-7B</td>
<td>Text to Code</td>
<td>3.5s</td>
<td>7GB</td>
</tr>
</tbody>
</table>
Python hot-reloading

- Great developer experience with runtime code change.
- Removes the need for reset at each iteration step.
- Contains fail-safe methods for avoiding big undefined behavior.
- Integrates well with Large Language Models (LLMs).
## Comparative Table

<table>
<thead>
<tr>
<th></th>
<th>dora-rs</th>
<th>ROS 2 (Humble)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tier 1 Support</strong></td>
<td>Python, Rust</td>
<td>C, C++</td>
</tr>
<tr>
<td><strong>Tier 2 Support</strong></td>
<td>C, C++, ROS2</td>
<td>Python, Rust</td>
</tr>
<tr>
<td><strong>Compilation</strong></td>
<td>No Compilation</td>
<td>Colcon</td>
</tr>
<tr>
<td><strong>Hot-reloading</strong></td>
<td>Python</td>
<td></td>
</tr>
<tr>
<td><strong>Message Format</strong></td>
<td>Arrow</td>
<td>ROS Message (Protobuf)</td>
</tr>
<tr>
<td><strong>Local Communication</strong></td>
<td>Shared Memory</td>
<td>Shared Memory for C/C++</td>
</tr>
<tr>
<td><strong>Remote Communication</strong></td>
<td>TCP (experimental)</td>
<td>DDS</td>
</tr>
<tr>
<td><strong>Metrics, Tracing, and Logging</strong></td>
<td>Opentelemetry</td>
<td><strong>ROS2 custom tooling</strong></td>
</tr>
<tr>
<td><strong>Data archives</strong></td>
<td>Parquet</td>
<td>Rosbag</td>
</tr>
<tr>
<td><strong>Supported Platforms</strong></td>
<td>Windows, macOS, Linux (All Platform)</td>
<td><strong>Windows 10, Ubuntu 22.04</strong></td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td>YAML</td>
<td>XML</td>
</tr>
</tbody>
</table>
Coming up!

Partnership with Huggingface that brings the help of:

- Thomas Wolf, cofounder of Hugging Face, one of the leading AI Startups
- Remi Cadene, Principal at HuggingFace, and ex-Tesla that used to lead the team that build Optimus
dora-arms: Powering robotics arms
dora-arms: Powering robotics arms

Promising result:

- Increase by 10x the frequency (from 50Hz to 500Hz) on teleoperation from SOTA aloha robotic arms, and reduced the latency to 2ms of communication between the 2 arms by having only a single process.
- Fine-grained teleoperation data for training as well as providing better control on teleop, which should lead to better dataset and models.
- Removed the need for complex installation and instead use cargo
- Making robotic cross-platform: Linux, MacOS, Windows and available on: Rust, C, C++, Python
Thanks for listening 😊

- [github.com/dora-rs/dora](https://github.com/dora-rs/dora) 🌟 1k
ROS2 Bridge

- Allows gradual migration of existing ROS2 applications
- Makes it possible to use ROS2 tooling with Dora

Current Work in Progress:

- Communication via DDS middleware
- Autogenerate Rust and C++ bindings for ROS2 message files.
- Automatic type conversions between ROS2 type, Arrow Type and Native Type
Opentelemetry

- Uses Opentelemetry for logs, tracing and metrics
- Language agnostic, Backend agnostic, and handles distributed systems
- Linking logs, traces and metrics with a same abstraction.
Hot Reloading for Python

- Enables code change at runtime keeping current states intact.
- Removes the need for reset at each iteration step.
- Contains fail-safe methods enabling to fail fast.