If you have done this at home, feel free to skip. Thank you!

Repository setup

Step 1: Clone repository (ONLINE method)

git clone --recursive https://github.com/ros-

realtime/roscon-2023-realtime-workshop.git code

Step 1: Clone repository (OFFLINE method)

- 1. Borrow a Raspberry Pi and connect directly via Ethernet
- 2. Download http://192.168.10.1/data/repository.tar.gz
- 3. Extract the tarball: tar xzf repository.tar.gz

Step 2: Import and start Docker container

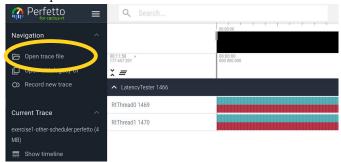
You need this for both laptop- and Raspberry-Pi-based workflows!

- 1. Borrow a Raspberry Pi and connect directly via Ethernet
- 2. cd into the downloaded repository
- 3. Run docker/fetch
- 4. Run docker/start
- Run docker/shell

Step 3: Test compiling and running exercise 1

Run all commands inside the Docker shell started above

- cd /code/exercise1
- 2. colcon build
- 3. ./run.sh
- 4. This should create a file called exercise1.perfetto
- 5. Go to http://localhost:3100
- 6. Click Open trace file on top left
- 7. Open the exercise1.perfetto file
- 8. Consult with Perfetto trace viewer guide
- 9. Expected result:



If compilation fails with missing dependencies, check that you cloned with the --recursive flag, or simply clone the repository using the offline method.

Laptop workflow

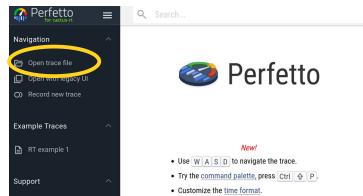
- 1. Edit the code in the repo with your preferred editor
- 2. cd to the repo in a terminal
- 3. Login to the Docker container via docker/shell
- 4. Inside the shell, cd /code/exercise1
- Compile and run the exercise according to instructions on slides and/or exercise README. This will generate a file named exercise<X>.perfetto in the same directory. Example: exercise1.perfetto
- 6. Go to http://localhost:3100 and Open trace file with the file above.
- 7. Go back to step 1.

Raspberry Pi workflow

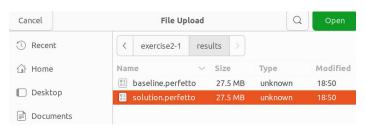
- 1. Connect Raspberry Pi directly to your laptop via Ethernet.
- 2. Edit the code in the repo with your preferred editor
- 3. Login to the Docker container via docker/shell
- 4. Run the command upload-to-pi
- 5. Login to the Raspberry Pi with:
 - 1. ssh <u>ubuntu@192.168.10.1</u>
 - 2. Password is <u>ubuntu</u>
- 6. After login, cd /code/exercise1
- Compile and run the exercise according to instructions on slides and/or exercise README. This will generate a file named exercise
 perfetto in the same directory.
- Download the trace file by browsing to <u>http://192.168.10.1/repo/</u> and clicking on the right perfetto trace file
- 9. Go to http://localhost:3100 and Open trace file with the downloaded trace file above
- 10. Go back to step 2.

Loading data in Perfetto

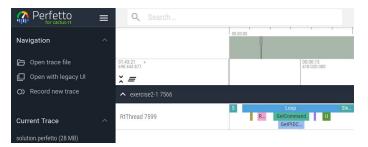
Step 1: A locally-hosted version of Perfetto is available with the Docker container, at http://localhost:3100. Go there and you will find the following interface:



Step 2: To open a trace file, click <u>Open trace file</u> and select a file:

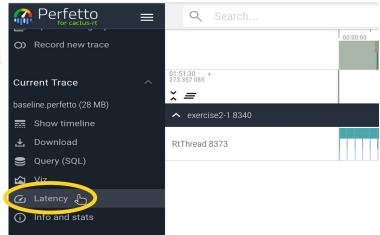


Step 3: Use **WASD** to navigate. **W**: zoom in; **S**: zoom out; **A**: pan left; **D**: pan right. Zoom and pan until you see the following:



Using histogram visualization

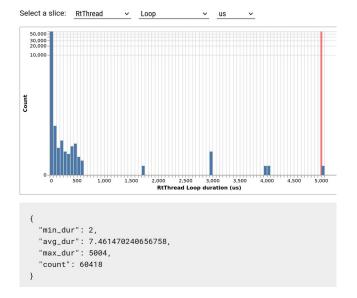
After loading data, on the left side bar, click on <u>Latency</u>:



Select a thread and a slice on the drop down:

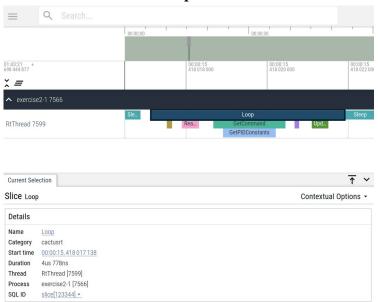


Visualize the latency histogram. Click on <u>us</u> to change the time scale if necessary. Min, average, max duration is also shown below. Red vertical line shows maximum latency:



Find the longest slice

Click on a slice such as **Loop**:



Click on the <u>Loop</u> link at the bottom then click <u>Slices</u> with same name in the popup menu



Clock on <u>Duration</u> in the table header then <u>Sort: highest</u> first:



Click on the the ID shown on the left most column (96282 in the above example) to bring the timeline view to the longest **Loop** instance:

