

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

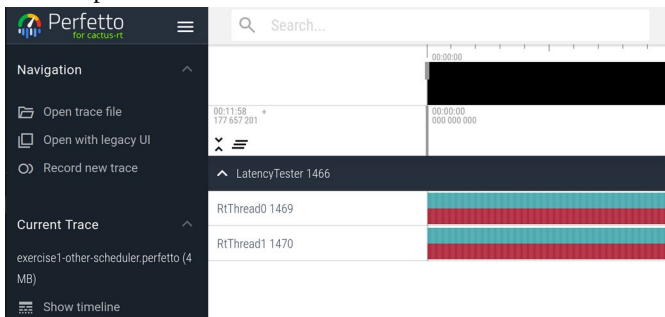
You need this for both laptop- and Raspberry-Pi-based workflows!

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

Step 3: Test compiling and running exercise 1

Run all commands inside the Docker shell started above

1. `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

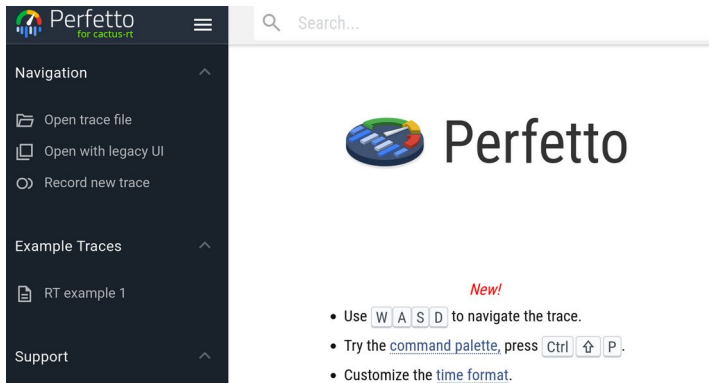
1. Edit the code in the repo with your preferred editor
2. `cd` to the repo in a terminal
3. Login to the Docker container via `docker/shell`
4. Inside the shell, `cd /code/exercise1`
5. Compile and run the exercise according to instructions on slides and/or exercise README. This will generate a file 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

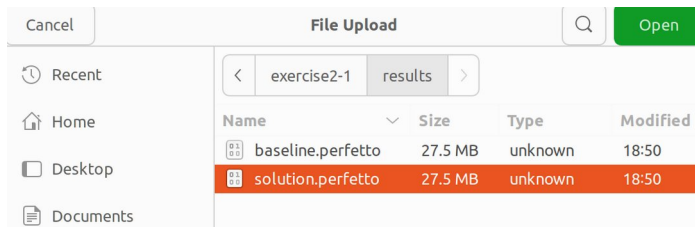
1. Connect Raspberry Pi directly to your laptop via Ethernet.
2. Edit the code in the repo with your preferred editor
3. Login to the Docker container via `docker/shell`
4. Run the command `upload-to-pi`
5. Login to the Raspberry Pi with:
 1. `ssh ubuntu@192.168.10.1`
 2. Password is `ubuntu`
6. After login, `cd /code/exercise1`
7. 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.
8. Download the trace file by browsing to <http://192.168.10.1/repo/> 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

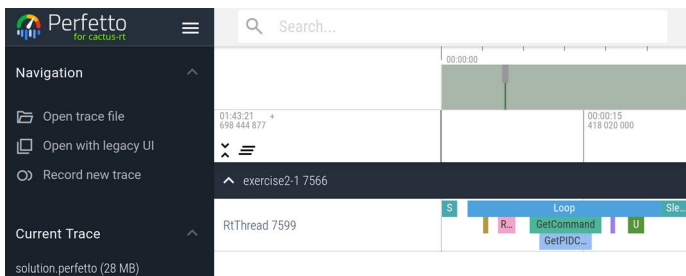
Step 1: A locally-hosted version of Perfetto is available with the Docker container, at <http://localhost:3100>. Go there and you will find the following interface:



Step 2: To open a trace file, click Open trace file and select a file:

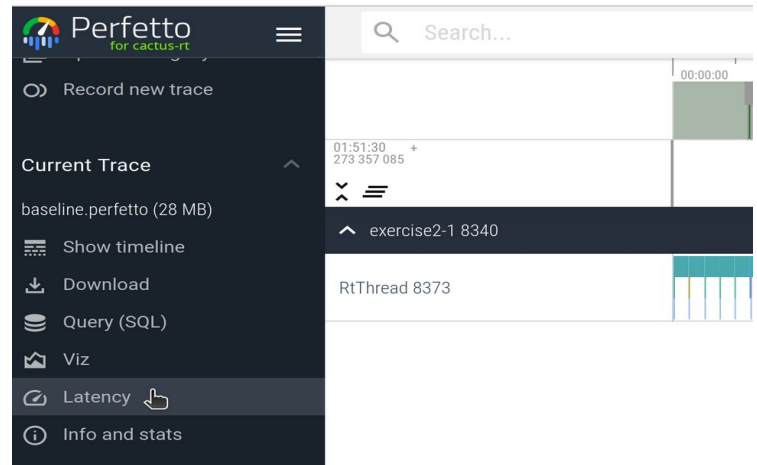


Step 3: Use **WASD** to navigate. **W**: zoom in; **S**: zoom out; **A**: pan left; **D**: pan right. Zoom and pan until you see the following:

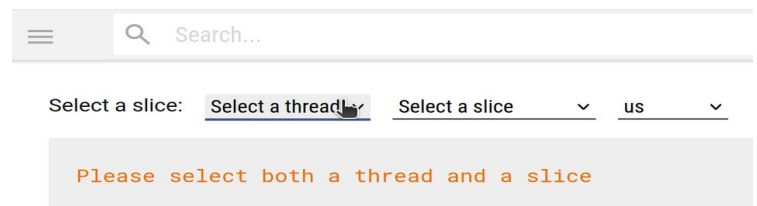


Using histogram visualization

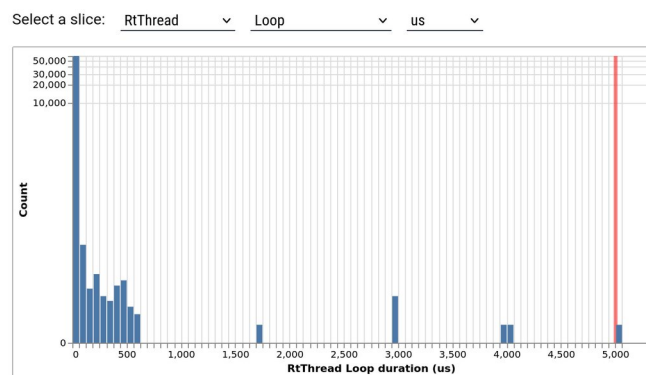
After loading data, on the left side bar, click on Latency:



Select a thread and a slice on the drop down:



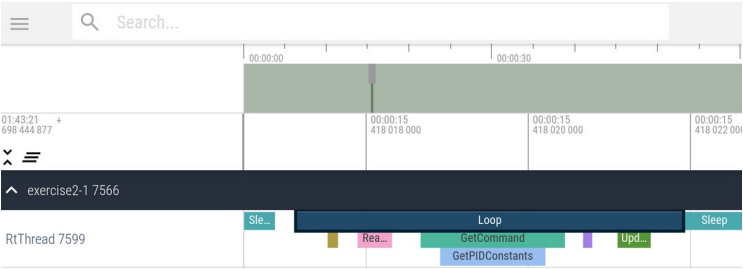
Visualize the latency histogram. Click on us to change the time scale if necessary. Min, average, max duration is also shown below. Red vertical line shows maximum latency:



```
{
  "min_dur": 2,
  "avg_dur": 7.461470240656758,
  "max_dur": 5004,
  "count": 60418
}
```

Find the longest slice

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



Current Selection

Slice Loop Contextual Options

Details

Name

Loop

Category

cactusr

Start time

00:00:15.418.017.138

Duration

4us 778ns

Thread

RtThread [7599]

Process

exercise2-1 [7566]

SQL ID

slice[123344]

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

Table slice Showing rows 1-100 of 60418 Show debug track Copy SQL query Close

X name = 'Loop'

ID	Timestamp	Duration	Thread duration	Category	Name	Thread name	tid	Process name	pid
0	00:00:00.000.000			cactusr	Loop	RtThread	8373	exercise2-1	8340
8	00:00:00.001.000			cactusr	Loop	RtThread	8373	exercise2-1	8340
16	00:00:00.002.000			cactusr	Loop	RtThread	8373	exercise2-1	8340

Click on Duration in the table header then Sort: highest first

Table slice Showing rows 1-100 of 60418 Show debug track Copy SQL query Close

X name = 'Loop'

ID	Timestamp	Duration	Thread duration	Category	Name	Thread name	tid	Process name	pid
96282	00:00:12.037.704.512	5ms 4us 944ns	NULL	cactusr	Loop	RtThread	8373	exercise2-1	8340
21520	00:00:26.916.704.818	4ms 42us 796ns	NULL	cactusr	Loop	RtThread	8373	exercise2-1	8340

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:

