



Mobile robotics scale-up leveraging ROS

Guillaume Doisy - Lead System Architect – @doisyg

Marcus Scheunemann – Head of Autonomy - @scheunemann

ROSCon 2024 – 23/10/2024

DEXORY



Context

ROS 2 developer experience at Dexory

Software integration

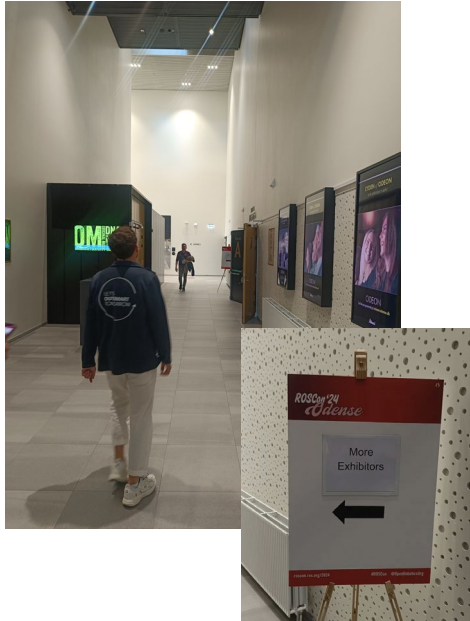
Bug hunting diary

Working with ROS open source software

DEXORY

Context

Dexory's AGV



Context

Dexory's AGV

- 600kg / 14 meters scanning autonomous mobile robot
- Up to 185 000 m2 (2M ft2!) warehouse, and counting...
- 1 robot per warehouse
- 6x 3D lidars + 2x 2D lidars
- First CAD in September 2022 – 15 people company
- First deployment in March 2023
- Today (October 2024):
 - 180+ people
 - Strong funding (80M Series B)
 - Over 90 units built in the last 12 months
 - Currently deploying 3 robots a week
 - Global customers:
 - UK
 - Pan EU
 - USA
 - MENA



ROS 2 developer experience at Dexory

DevOps

- Get DevOps resources and CI/CD running early
- Standardized developer environment (docker or other), no more “it runs on my machine”. With some flexibility (e.g. not only VS Code)
- Linting (pre-compilation)
- (fast) CI build check
- (fast) CD binaries generation
- Dependency control
- Allow non-persistent production image alterations (launch files, parameters files, packages)

Pro tip: visit “Session 1a - ROS testing and tools” at 13:50 with Ruffin and Marcus.

Accelerating the CI/CD-to-robot cycle by 10x for 1/10th the cost



The screenshot shows a GitHub Actions workflow run for the pull request 'Add tf timeout #345'. The workflow is titled 'Build ARRI production image' and is currently in a 'Call' step. The 'Call' step is expanded to show a list of jobs: 'Feature / Build Base Images' and 'Feature / Colcon / Build Worksp...'. The 'build / style' job is selected and expanded to show its steps: 'Set up job', 'Initialize containers', 'Run actions/checkout@v3', 'Run andstor/file-existence-action@v2', 'Run actions/setup-python@v4', 'Run botsandus/ament_lint_pre_commit@v0.0.9', 'Post Run botsandus/ament_lint_pre_commit@v0.0.9', 'Post Run actions/setup-python@v4', 'Post Run actions/checkout@v3', 'Stop containers', and 'Complete job'. The workflow is shown as successful, having completed 22 minutes ago in 41s.

ROS 2 developer experience at Dexory

Simulation

- All sensors and actuators are worth simulating, even roughly (e.g. charging behavior)
- Test sensor processing with bags
- API pipeline and behaviors with a simulation environment
- RTF \geq 1.0 running on isolated developers machines. With 8 lidars thanks to the Gazebo Robotec.ai RGL plugin.
- For mobile robots, 2.5D simulation: topological simulation (i.e. customer site reproduction)



<https://github.com/RobotecAI/RGLGazeboPlugin>
<https://github.com/RobotecAI/RobotecGPULidar>



ROS 2 developer experience at Dexory

Tooling

- Make ssh easy for developers, leverage VPN / Wireguard / Tailscale
- Easy ros bag recording and downloading.
- Record raw, replay with filter and sensor pipeline launch files (SIMULATION / use_sim_time).
- Live visualization of data. Don't reinvent the wheel: Rviz + VNC
- Logs, logs all the time and everywhere (with Grafana / Telegraf / Prometheus).



ROS 2 developer experience at Dexory

Keep it simple, test continuously test, deploy early

- Developing new features in a RaaS scale-up is (also) **hard**
- specifically when you need to keep existing customers happy

- Challenge:
 - each new feature increases complexity!
 - thus each change increases efforts for later features

Solution:

- Keep features minimal, focus only customer needs
- deploy early

Pro-tip:

- Do not call it testing (it's boring; it feels like nothing is done)
- link it to increasing “annual contract value” of a customer subscription (OKR); coin it as “decrease feature time to market” (KPI)
- Get people on testing (test engineer; QAs)



Software integration

Packaging

- Tried historically (pre-Dexory):
 - Ubuntu Desktop with building from source (don't do that)
 - Ubuntu Desktop with CI generated debian packages (blooming and versioning hell)
 - Ubuntu Desktop with Docker (but Ubuntu Desktop is not an IoT distrib, boot will be destroyed at some point)
 - Ubuntu Core with snap (cleanest approach so far but... snaps)
 - Yocto
- Today:



The screenshot shows the Balena Cloud IoT dashboard. The top navigation bar includes 'Getting Started', 'Docs', 'Roadmap', 'Forums', 'Status', 'balenaHub', and the user 'Guillaume Dolsy'. The left sidebar contains navigation options: Organizations, BotsAndUs, Fleets, arri, Devices, arri-10, Summary, Device Variables, Device Configuration, Actions, Settings, Diagnostics, and Location.

Service	Status	Release
camera-manager	Running	1.0.1929
capture	Running	1.0.1929
firmware-manager	Running	1.0.1929
kernel-modules	Running	1.0.1929
litestream	Running	1.0.1929
logging	Running	1.0.1929
metrics	Running	1.0.1929
mosaic-base	Running	1.0.1929
networking	Running	1.0.1929
pulseaudio	Running	1.0.1929
remote-desktop	Running	1.0.1929
robot-manager	Running	1.0.1929
ros	Exited	1.0.1929
ros_prod	Running	1.0.1929
telegraf	Running	1.0.1929

Below the table, there are system resource monitors: CPU (30%), Temperature (87C), Memory (8.1 GB / 15.3 GB), and Storage (37.0 GB / 230.9 GB). A 'Logs' section is visible at the bottom with options for UTC and Timestamps.

Software integration

Controlling external dependencies

- Fix everything: binaries, sdk, sensor driver, etc...
- Why ?
 - Trust open source with control
 - Upstream can disappear
 - Uncontrolled updates and regressions
 - Independency of the build chain
- From source: fork
- From binaries: mirror repos, classically Ubuntu + ROS with aptly

Accidental Iron sync, 2025-05-23

■ Packaging and Release Management ■ Iron ros2, release, iron, sync



marcogg Great contributor

25d

An accidental Iron sync went out at 🌐 [May 23, 2024 2:09 PM](#) that left an inconsistent state on the main repositories for Iron. We would like to apologize for that. This has been mitigated and a second sync went in with the necessary fixes at 🌐 [May 23, 2024 3:37 PM](#) .

Thanks for your understanding.

Software integration

Taming DDS

- Biggest ROS 2 entry pain point
- Some complains:
 - Very not optimized default settings for my standard use-case
 - Cross-talk by default
 - One drama per ROS update
- Our Iron recipe:

```
export RMW_IMPLEMENTATION=rmw_cyclonedds_cpp
export ROS_AUTOMATIC_DISCOVERY_RANGE=SUBNET
# Or, if no need for network com:
# export ROS_AUTOMATIC_DISCOVERY_RANGE=LOCALHOST
export CYCLONEDDS_URI=${CYCLONEDDS_URI:-
"<CycloneDDS><Domain><General><Interfaces><NetworkInterface
name=\"lo\"/></Interfaces><AllowMulticast>true</AllowMulticast></Gen
eral><Discovery><ParticipantIndex>none</ParticipantIndex></Discovery
></Domain></CycloneDDS><Gen><Allow>spdp</Allow></Gen>"}

```

```
ip link set lo multicast on
sysctl -w net.core.rmem_max=2147483647
```



Software integration

Time synchronization

- Choose sensors with PTP support (if you can)
- Use your computer network interface as PTP hardware grandmaster (no need for fancy equipment)
- Synchronize hardware clock with system clock:

```
echo "Starting ptp4l and phc2sys for PTP provision..."  
/usr/sbin/ptp4l -f /etc/linuxptp/ptp4l.conf boundary_clock_jbod=1 clockClass=128 -i ${WIRED_IFACE} &  
/usr/sbin/phc2sys -w -s CLOCK_REALTIME -c ${WIRED_IFACE} &
```

- Synchronize system clock with world time with NTP (chrony)

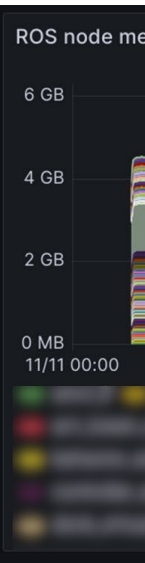
Bug hunting diary



Bug hunting diary

Elusive memory leak

- Only appears after a specific dock sequence
- Hunted with Grafana
- Publishing outdated TF at high frequency is increasing memory usage on all TF listeners
- Fix merged:
<https://github.com/ros2/geometry2/pull/636>
(thanks Alejandro)



A screenshot of a GitHub pull request for the repository 'ros2 / geometry2'. The pull request title is 'Fix constantly increasing memory in std::list #636'. It is marked as 'Merged' and was merged by 'ahcorde' on Jan 10. The pull request description includes a 'Description' section explaining the issue: 'When someone is constantly publishing with the same tf2 timestamp (application error, I know), the storage_ of the tf2::TimeCache grows unbounded causing system-wide memory leaks. In a ROS system, each tf2 Listener will spawn one of these TimeCache objects and thus, all ROS nodes that have any sort of tf2 listener will slowly start allocating more and more memory'. It also includes a 'How to test it' section with instructions on how to reproduce the issue using a small repo. The right sidebar shows the 'Reviewers' section with 'clalancette' and 'ahcorde' listed, and the 'Assignees' section with 'clalancette' and 'ahcorde' listed. The 'Labels' section is empty, and the 'Projects' section is also empty. The 'Milestone' section is empty. The 'Development' section shows a message: 'Successfully merging this pull request may close these issues.'

Bug hunting diary

Intel instruction set fun

- Binaries built locally work / Binaries built by CI crash at runtime
- Works on NUC11 / Crashes on NUC13
- Similar nav2 story



ros-navigation / navigation2

<> Code Issues 77 Pull requests 26 Discussions Actions

MPPI crashing on loading plug-ins #3767

Closed AmmarAlbakri opened this issue on Aug 21, 2023 · 47 comments

AmmarAlbakri commented on Aug 21, 2023

Bug report

- Operating System:
 - Ubuntu 22.04
- ROS2 Version:
- Humble Source

After the binary packages sync on 18.08.2023 the `nav2_mppi_controller` started crashing on when launching;

```
[controller_server-1] [INFO] [1692618734.372369699] [controller_server]: Created progress_checker : progress_checker of type nav2_controller::ProgressChecker
[controller_server-1] [INFO] [1692618734.373032712] [controller_server]: Created goal_checker : goal_checker of type nav2_controller::SimpleGoalChecker
[controller_server-1] [INFO] [1692618734.373172709] [controller_server]: Controller Server has goal_checker, goal checkers available.
[controller_server-1] [INFO] [1692618734.376989968] [controller_server]: Created controller : FollowPath of type nav2_rotation_skin_controller::FollowPathController
[controller_server-1] [INFO] [1692618734.378278494] [controller_server]: Created internal controller for rotation shimming: FollowPath of type nav2_mppi_controller::MPPIController
[controller_server-1] [WARN] [1692618734.378919044] [controller_server]: Controller period is less than model dt, consider setting it eq
[controller_server-1] [INFO] [1692618734.380392088] [controller_server]: ConstraintL1tic instantiated with 1 power and 4.000000 weight.
[controller_server-1] [INFO] [1692618734.380438022] [controller_server]: Critic loaded : app::critic::ConstraintCritic
[ERROR] [controller_server-1]: process has died [pid 92919, exit code -4, cmd '/opt/ros/humble/lib/nav2_controller/controller_server --ros-args --param-file /tmp/bugbash0n --freqHz 10 --tf_static:=tf_static /cmd_vel:=cmd_vel_nav --odom:=diffdrive_controller/odom']
```

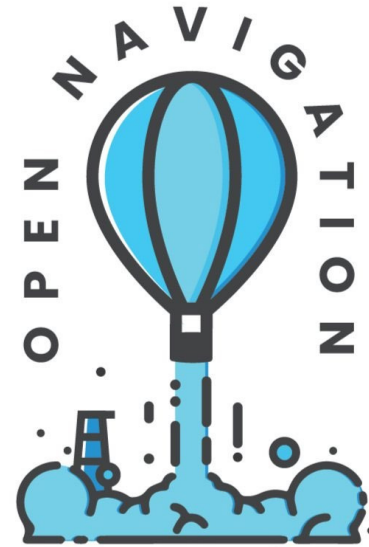
ps: same non-updated setup still working.

By the way, the sync was supposed to fix this issue but it hasn't been solved yet: [#3762 \(comment\)](#)

<https://github.com/ros-navigation/navigation2/issues/3767>

Working with ROS open source software

- Advantages and drawbacks of using an updated version of ROS 2
- Value of open source collaboration for Dexory
- Feedback of 1.5+ years of sponsoring Open Navigation / nav2
- Concerns about the maintenance of the core ROS packages / OSRA future





Thanks !

(we are recruiting)

Guillaume Doisy - Lead System Architect – @doisyg

Marcus Scheunemann – Head of Autonomy - @scheunemann

ROSCon 2024 – 23/10/2024

DEXORY