Are all simulations just an exercise in hand waving?

Testing real hardware capabilities
with a simulator shouldn’t be hard

You ever run a simulation and everything just worked?

But then you ran it on your platform and didn’t get so lucky…

- Powerful computers running a simulation with adaptive clock and idealized sensors noise models only gives a fraction of the true systems constraints.

- Compute bottlenecks have been and will probably always be a factor for any fielded system, it is a fundamental optimization problem around SWaP-C2 that cannot be ignored. Real robots eventually graduate past “strap a server on it” MVP.

- How do you even begin to identify bottlenecks on a real platform from a simulation?

Time synchronized hardware logging

ros2_psutil:

- Match data logging from sensors to system usage and performance logging

- Non-intrusive and highly optimized for specific constraint concerns

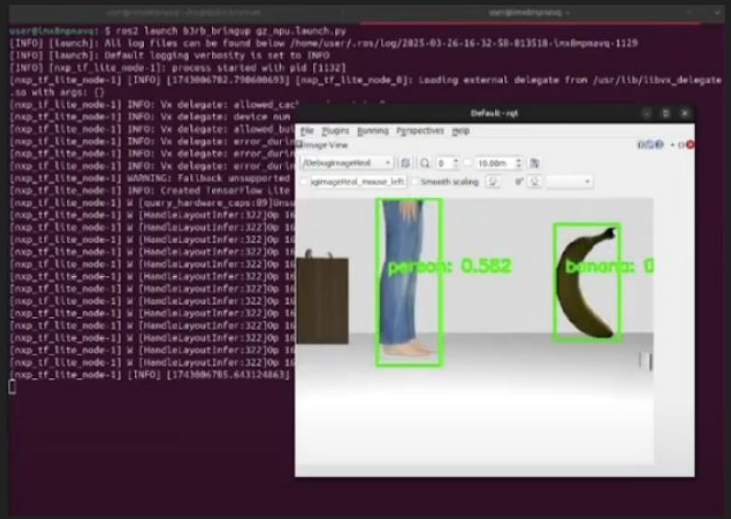
```
'update_frequency':
    Update frequency to publish in Hz.
    (default: '1.0')

'individual_topics':
    Publish individual topics instead of unified psutil topic.
    (default: 'False')

'mem':
    Publish memory.
    (default: 'True')

'net':
    Publish network.
    (default: 'True')

'net_addr':
    Publish network address.
    (default: 'True')

'net_state':
    Publish network state.
    (default: 'True')

'net_stats':
    Publish network stats.
    (default: 'True')

'temp':
    Publish temperature.
    (default: 'True')

'proc':
    Publish processor.
    (default: 'True')

'proc_per':
    Publish processor percent.
    (default: 'True')

'proc_freq':
    Publish processor frequency.
    (default: 'True')

'net_nic_match':
    Array of network NICs to return, to return all leave empty.
    (default: '[""]')

'net_af_match':
    Int array of network address families to return, valid values are:
        AF_INET: 2,
        AF_INET6: 10,
        AF_PACKET: 17,
        to return all leave empty.
    (default: '[-9999]')

'dev_temp_match':
    Array of devices to return temperatures for, to return all leave empty.
    (default: '[""]')

'temp_name_match':
    Array of temperture names on a device to return, to return all leave empty.
    (default: '[""]')
```

Leverage GZ transport where possible

Sending an image over ros_gz_bridge is not an ideal solution, you have to go from a local zmq subscriber that translates PB to CDR and sends it back out over DDS.

Instead subscribe directly to gz-transport on your platform (yes this works for arm64 based systems as well)

- GZ_IP=<xxx.xxx.xxx.xxx>
- GZ_PARTITION=<named_relay>
- GZ_RELAY=<xxx.xxx.xxx.xxx>



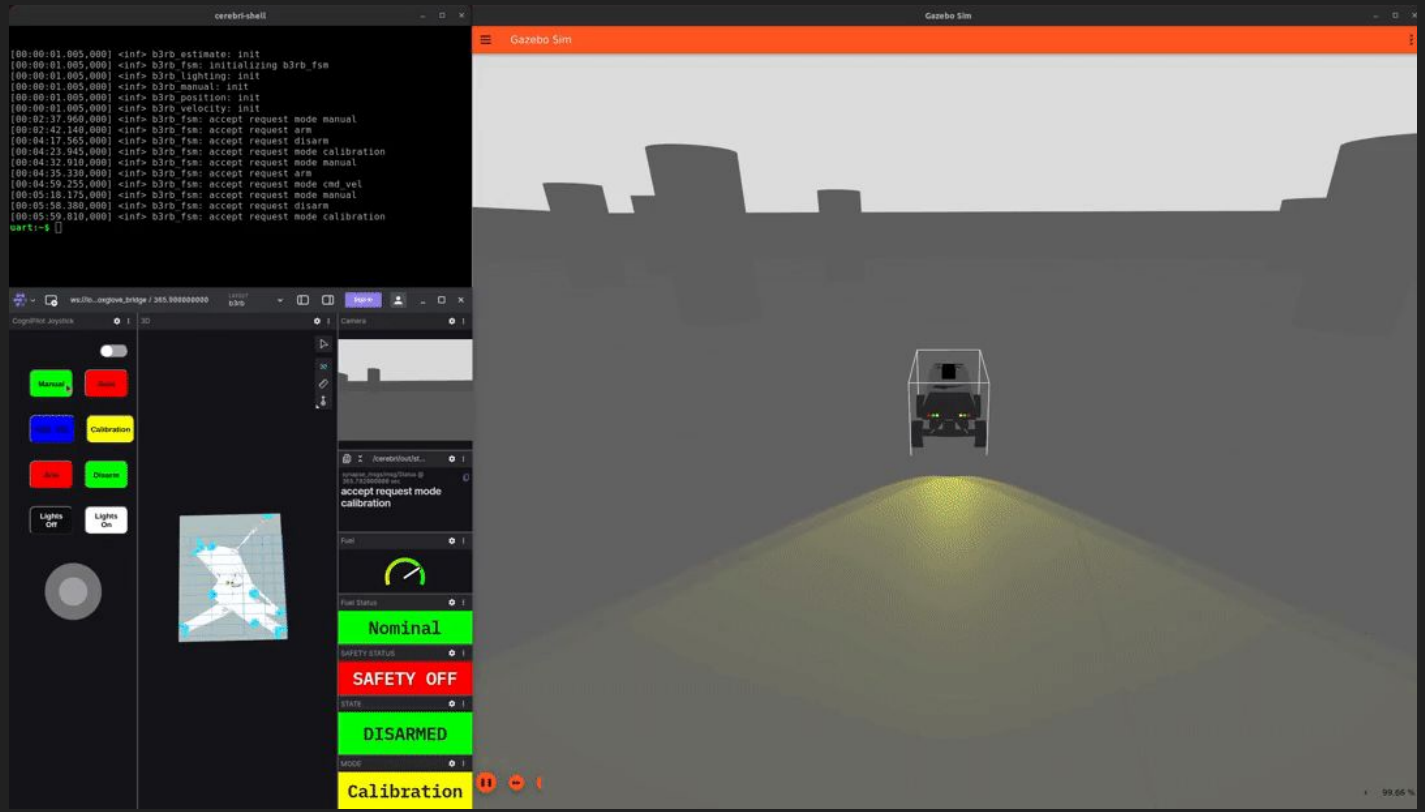
Check your gz_transport topic frequency

Use the CLI to see if you are having network related issues, minimize network overhead where possible.

```
user@user:~$ gz topic -l
/actuators
/camera/camera_info
/camera/image_raw
/clock
/gazebo/resource_paths
/gui/camera/pose
/gui/currently_tracked
/gui/track
/model/b3rb/battery/linear_battery/state
/model/b3rb/odometry
/model/b3rb/odometry_with_covariance
/model/b3rb/pose
```

```
user@user:~$ gz topic -f -t /camera/image_raw

interval [0]:    0.0304533s
interval [1]:    0.0328695s
interval [2]:    0.0369859s
interval [3]:    0.032835s
interval [4]:    0.0303573s
interval [5]:    0.0331047s
interval [6]:    0.0363206s
interval [7]:    0.0303834s
interval [8]:    0.033276s
interval [9]:    0.030912s
average rate: 30.5346
min: 0.0303573s max: 0.0369859s std dev: 0.00226044s window: 10
```

Match your full system - scene luminance matters



gz_transport15 has gone Zenoh!

Now you can use it directly in ROS without the extra conversion pains of a translation bridge.

Some other planned additions to gazebo:

- Dynamic sensor noise models (“break” the sensors, not the simulation trying)

- Dynamic topic scaling (change the ODR of a sensor or data stream while it’s running)

Don’t forget to test Jetty!

It has some really great additions beyond Zenoh, one of our favorites is the auto-inertial.

LEADER BOARD

Place	User	Points
1	Creator-1705	579.0
2	akky20	382.0
3	nikodemj9	316.0
4	jmackay2	257.0
5	Physic69	236.0
6	jasmeet0915	151.2
7	avanmalleghem	120.0
8	Narashima1808	111.0
9	matosinho	84.4
10	SuperGops7	76.0