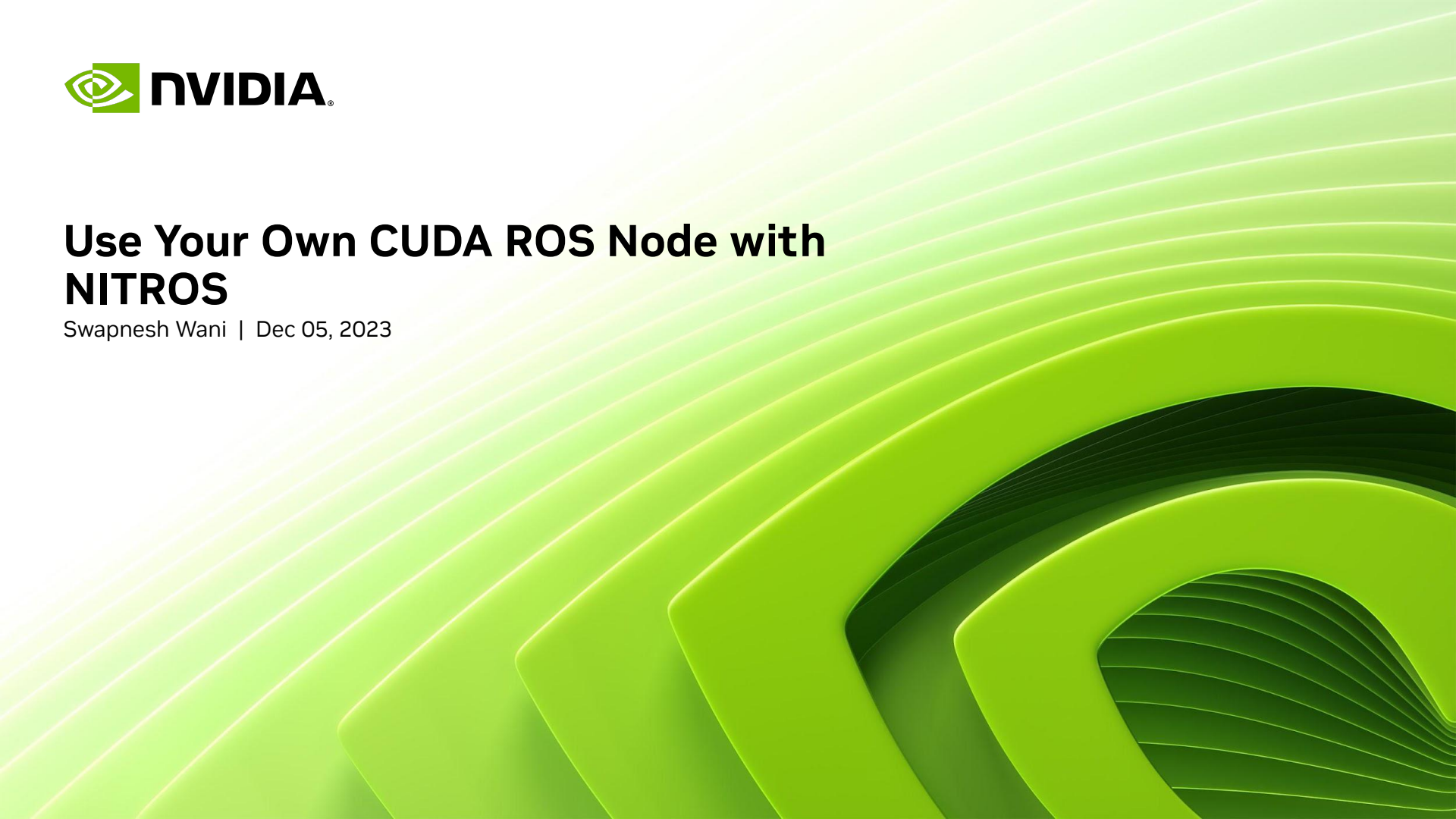


Use Your Own CUDA ROS Node with NITROS

Swapnesh Wani | Dec 05, 2023

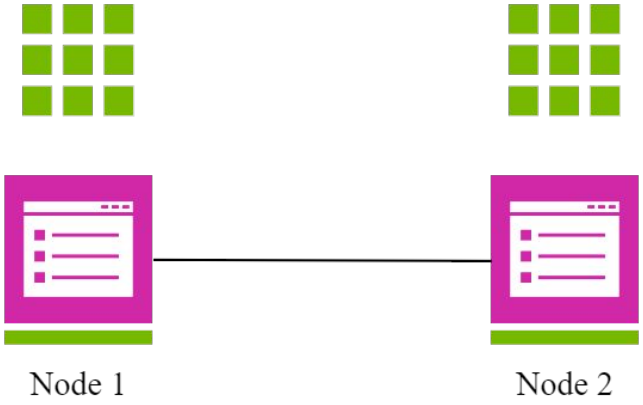
Agenda

- Overview
- NITROS
- Performance
- Managed NITROS APIs
- Demo examples
- Questions

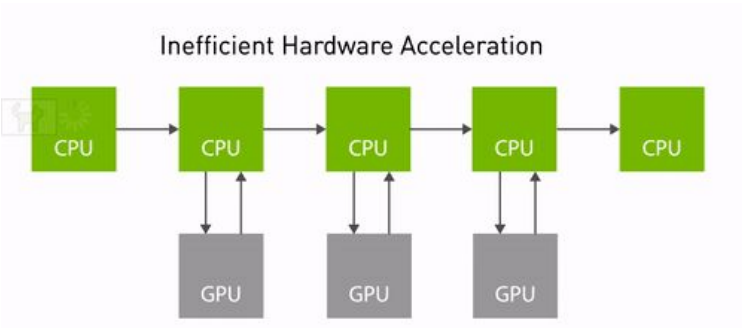
Motivation

Make efficient use of CUDA in ROS2 nodes

- **Scenario** - You have an awesome CUDA code that is part of ROS 2 nodes and want to share the GPU buffer between them.

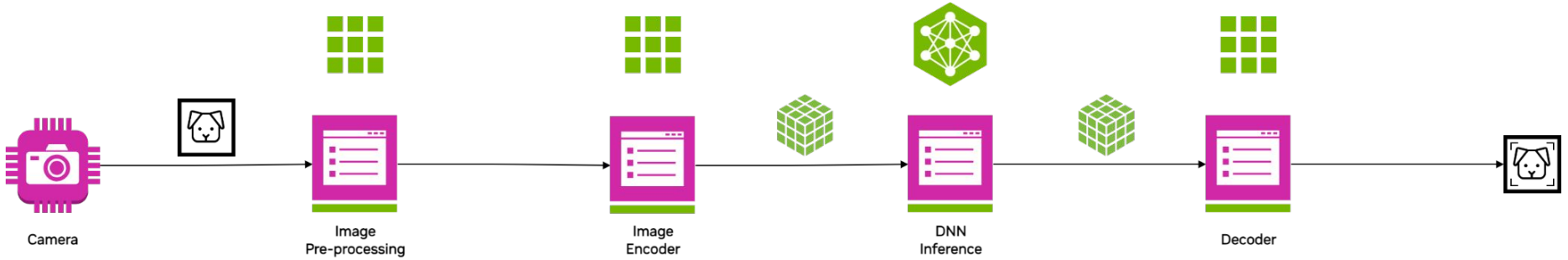


- **Problem** - Excessive memory copy between the two node.



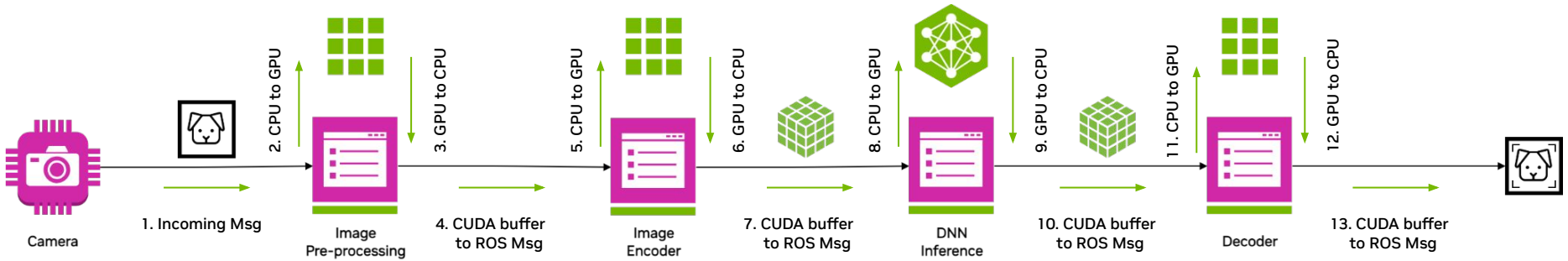
Real World Motivation

DNN inference pipeline



Real World Motivation

DNN inference details



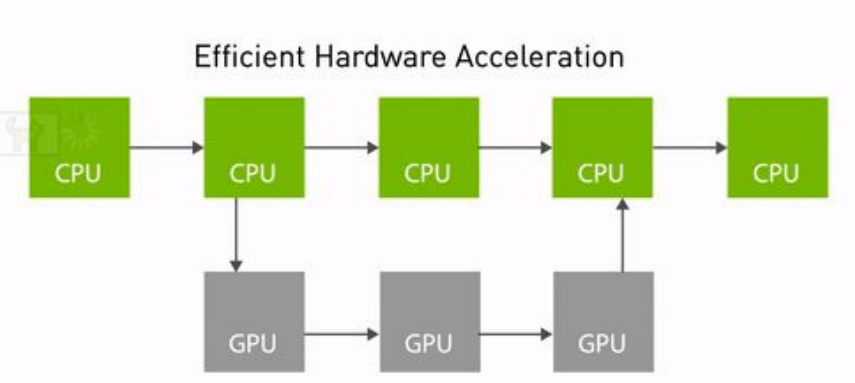
Resolution	Image Size	Extra Mem Copy
1280x720	~3 MB	~18 MB
1920x1080	~6.5 MB	~39 MB

6x penalty!

NITROS

Make efficient use of CUDA in ROS2 nodes

Solution



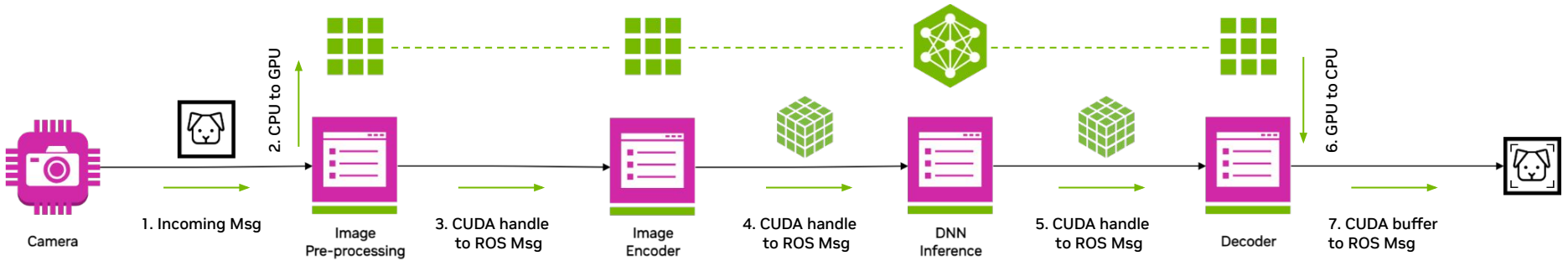
Isaac ROS NITROS

Benefits

- ✔ Efficient use of hardware
- ✔ Reduced memory copies
- ✔ Compatible with existing ROS 2 packages and tools
  - RViz / Foxglove
  - CLI tools like `ros2 topic echo <topic_name>`
  - Nav2, Moveit2, etc.

Real World Motivation

Now with NITROS



Resolution	Image Size	Extra Mem Copy
1280x720	~3 MB	0 MB
1920x1080	~6.5 MB	0 MB

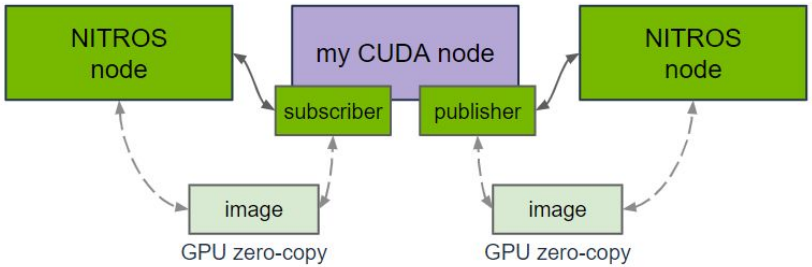
More throughput and less CPU load!

Performance

Of NITROS acceletrated graphs

Graph	Input Size	AGX Orin	AGX Xavier	Orin NX	Orin Nano 8GB	x86_64 w/ RTX 4060 Ti
AprilTag Graph	720p	213 fps 14 ms	122 fps 21 ms	102 fps 20 ms	71.0 fps 26 ms	472 fps 9.5 ms
Freespace Segmentation Graph	576p	45.9 fps 41 ms	19.2 fps 130 ms	27.6 fps 95 ms	21.3 fps 110 ms	91.0 fps 30 ms
Centerpose Pose Estimation Graph	VGA	36.1 fps 5.7 ms	20.2 fps 16 ms	19.4 fps 7.4 ms	13.8 fps 12 ms	50.2 fps 14 ms
DOPE Pose Estimation Graph	VGA	39.8 fps 33 ms	12.5 fps 160 ms	17.3 fps 120 ms	–	89.2 fps 15 ms
DNN Stereo Disparity Graph Full	576p	74.0 fps 20 ms	21.0 fps 93 ms	26.1 fps 42 ms	–	191 fps 11 ms
DNN Stereo Disparity Graph Light	288p	260 fps 13 ms	101 fps 20 ms	116 fps 16 ms	–	350 fps 12 ms
Stereo Disparity Graph	1080p	155 fps 15 ms	84.7 fps 21 ms	74.7 fps 21 ms	49.0 fps 31 ms	357 fps 3.0 ms
DetectNet Object Detection Graph	544p	232 fps 11 ms	90.0 fps 19 ms	105 fps 15 ms	74.2 fps 22 ms	644 fps 5.6 ms
TensorRT Graph PeopleSemSegNet	544p	421 fps 8.1 ms	252 fps 12 ms	238 fps 9.6 ms	162 fps 13 ms	704 fps 5.5 ms

CUDA with NITROS



- ✔ Ability to share GPU buffer across NITROS-enabled Isaac ROS node without extra copy
- ✔ Enabled increased parallel compute between CPU and GPU
- ✔ Simple APIs to use the NITROS publishers and subscribers.

Managed NITROS Publisher

- Provides interface for publishing messages in NITROS-enabled ROS graph.
- API is comparable to standard `rclcpp::Publisher`

```
// Declaring NITROS publisher

using namespace nvidia::isaac_ros::nitros;
std::shared_ptr<ManagedNitrosPublisher<
                  NitrosTensorList>> nitros_pub_;


// Creating NITROS publisher

nitros_pub_ = std::make_shared<
                  ManagedNitrosPublisher<
                  NitrosTensorList>>
                  (this,"my_topic_name",
                  nitros_tensor_list_nchw_rgb_f32_t::
                  supported_type_name);
```

Managed NITROS Subscriber

- Provides interface for subscribing messages from NITROS enabled ROS graph.
- API is comparable to standard `rclcpp::Subscriber`

```
// Declaring NITROS subscriber

using namespace nvidia::isaac_ros::nitros;

std::shared_ptr<ManagedNitrosSubscriber<
                  NitrosTensorListView>> nitros_sub_;


// Creating NITROS subscriber

nitros_sub_ = std::make_shared<
                  ManagedNitrosSubscriber<
                  NitrosTensorListView>>(
                  this, "my_topic_name",
                  nitros_tensor_list_nchw_rgb_f32_t::
                  supported_type_name,
                  std::bind(&MyDecoderNode::Callback,
                  This, std::placeholders::_1))
```

NITROS Builder

- Provides utility classes to create NITROS-typed messages.
- Classes offer builder-style interface which allows users to specify relevant fields during object creation one at a time.

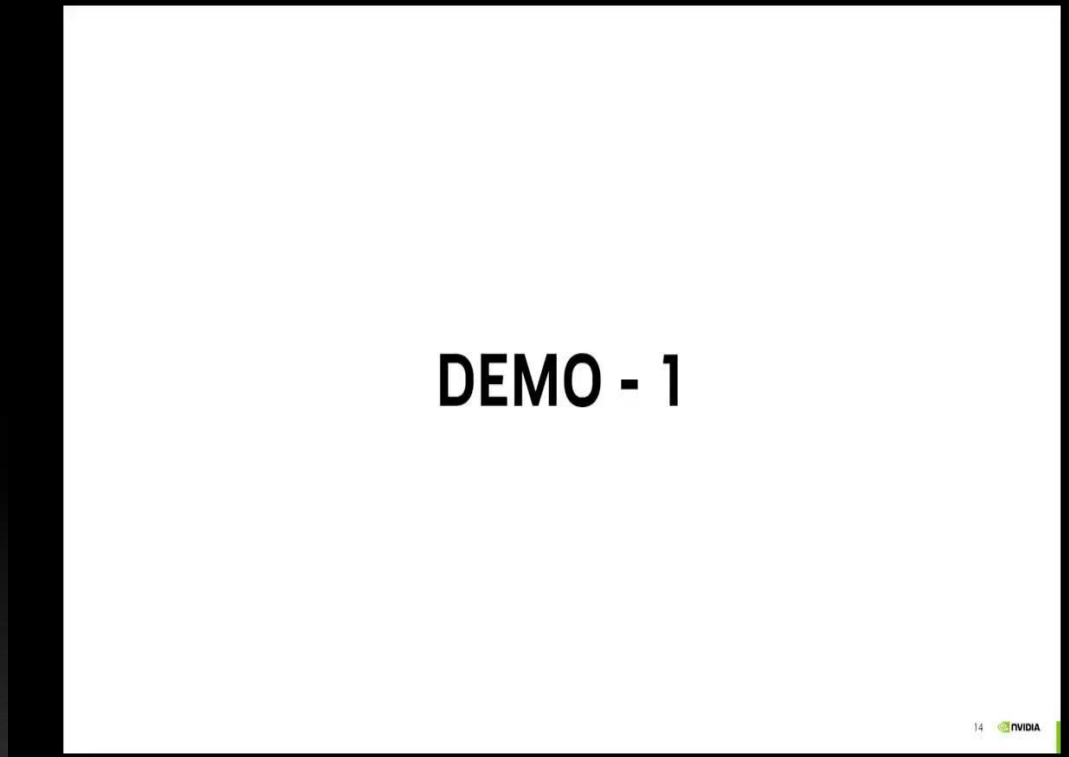
```
// NITROS builder example

using namespace nvidia::isaac_ros::nitros;
NitrosTensorList tensor_list =
                        NitrosTensorListBuilder()
                        .WithHeader(ros_header)
                        .AddTensor("tensor_1",
                                   foo_tensor)
                        .AddTensor("tensor_2",
                                   bar_tensor)
                        .Build();
```

NITROS Viewer

- Provides utility classes to access NITROS-typed messages.
- Classes offer methods to access relevant fields from a GPU backed buffer which can be copied back for further processing.

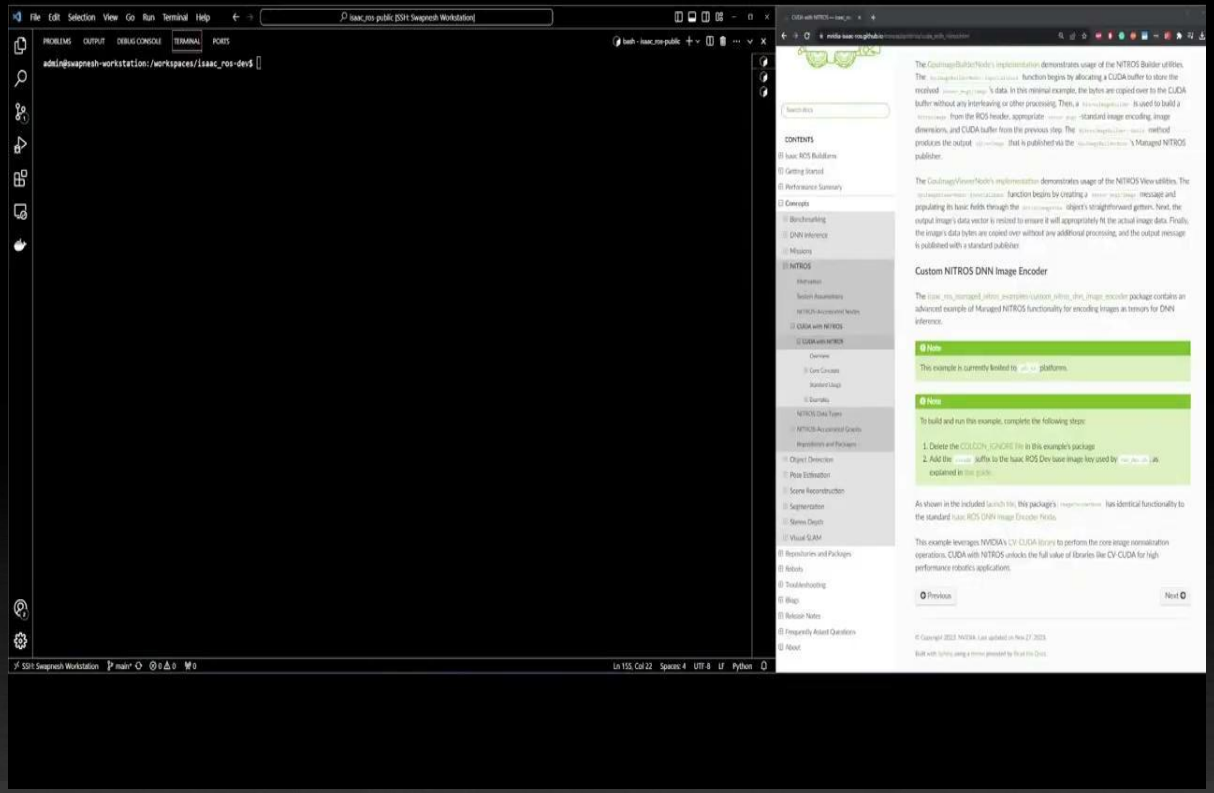
```
// NITROS viewer example

using namespace nvidia::isaac_ros::nitros;


void MyDecoderNode::Callback(const
NitrosTensorListView & msg)
{
  auto tensor = msg.GetNamedTensor(tensor_name_);
  size_t buffer_size{tensor.GetTensorSize()};
  std::vector<float> results_vector{};
  results_vector.resize(buffer_size);
  cudaMemcpy(results_vector.data(),
             tensor.GetBuffer(), buffer_size,
             cudaMemcpyDefault);
}
```

DEMO - 1



Demo 1

DEMO - 2


File Edit Selection View Go Run Terminal Help
isaac_ros-public [SSH: Swapnesh Workstation]
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
admin@swapnesh-workstation:/workspaces/isaac_ros-dev$
SSH: Swapnesh Workstation
Ln 155, Col 22 Spaces: 4 UTF-8 LF Python
CONTENTS
Custom NITROS DNN Image Encoder
Note
This example is currently limited to
New
To build and run this example, complete the following steps:
1. Delete the COLCON_IGNORE file in this example's package
2. Add the
suffix to the Isaac ROS Dev base image key used by
as explained in
As shown in the included
launch file, this package's
has identical functionality to the standard
This example leverages NVIDIA's
to perform the core image normalization operations. CUDA with NITROS unlocks the full value of libraries like CV-CUDA for high performance robotics applications.
Previous
Next

Demo 2

Conclusion

What did we learn

- ✔ How to efficiently use CUDA and integrate it with existing NITROS-enabled ROS graph
- ✔ How excessive memory copy can introduce bubbles in a hardware pipeline and affects the throughput and how to avoid it
- ✔ NITROS APIs and how to use them


NVIDIA
Questions?

Relevant links

Nvidia Isaac ROS Documentation Home

Performance Summary Page

Isaac ROS NITROS

CUDA with NITROS

Isaac ROS Webinar Series

NVIDIA Developer Blogs

ROS Discourse