Cloud Based Neural Net Inference

Topic 11

Main Idea of the Project

- WITH THE EMERGENCE IN AV, THERE'S AN INHERENT NEED TO PROCESS SENSOR DATA (MODEL INFERENCE IN PARTICULAR)
- DATA INFERENCE COULD BE EITHER DONE ON THE EDGE (LOCALHOST) OR ON THE CLOUD.
- TWO BROAD METHODOLOGIES TO CONNECT THE LOCALHOST AND CLOUD AND PROCESS: MQTT-BASED AND TENSORFLOW-SERVING BASED
- INTEGRATED USE OF ROSBAGS IN BOTH TFX AND MQTT. ALSO TFX RUNS COMPLETELY WITHOUT ROS.

Models Used

- **MODEL 1 SMALL : MOBILENET_V3_SMALL_968_608_OS8.PB**
  4.6 MB Frozen Graph Model (Input Width: 968, Input Height: 608)
- **MODEL 1 LARGE: MOBILENET_V3_LARGE_968_608_OS8.PB**
  8.6 MB Frozen Graph Model (Input Width: 968, Input Height: 608)
- **MODEL 2: MOBILENETV3_LARGE_OS8_DEEPLABV3PLUS_72MIOU**
  49.7 MB Saved Model (Width: 2048, Height: 1024)
- **MODEL 3: BEST_WEIGHTS_E=00231_VAL_LOSS=0.1518**
  29.4 MB Saved Model (Width: 2048, Height: 1024)

Communication Protocol

- **REST API**

  Communicates using JSON Payloads and HTTP/1.1 for transport. Performance delays due to JSON

- **GRPC (GOOGLE REMOTE PROCEDURE CALL)**

  Communicates uses HTTP/2 for transport and protobufs for serialization. Much Faster.

- **VERDICT**

  If you need maximum performance and are building systems with support for gRPC and Protocol Buffers, then gRPC is the better choice.

  If you prioritize simplicity, broad client support, and easy integrations, then the HTTP/REST API might be more suitable.

System Used

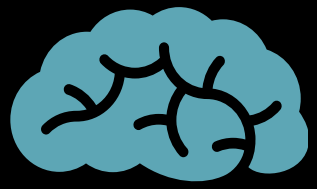
PC

- CPU Processors: 8 × Intel® Core™ i7-4720HQ CPU @ 2.60GHz
- RAM Memory: 15,5 GiB of RAM
- GPU: NVIDIA GeForce GTX 850M 4GB (Utilsed 3.7 GB)
- OS: Kubuntu 20.04
- OS Type: 64-bit
- TF Version (TFX) CPU latest (2.13)
- TF Version (TFX) GPU 2.11.1
- TF Version (MQTT) CPU latest (2.13)
- TF Version (MQTT) GPU 2.10.0

WS

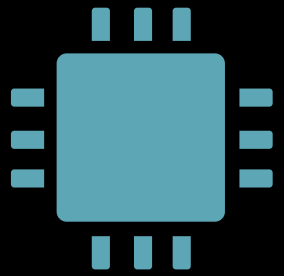
- CPU Procesors: 24 x AMD Ryzen Threadripper 3960X @ 3.8GHz
- RAM Memory: 62 GiB of RAM
- GPU: Nvidia RTX 3090 24GB (Utilised 23.4GB)
- OS: Ubuntu 20.04
- OS Type: 64-bit
- TF Version (TFX) CPU latest (2.13)
- TF Version (TFX) GPU 2.11.1
- TF Version (MQTT) CPU latest (2.13)
- TF Version (MQTT) GPU 2.10.0

MQTT-Based NN Inferencing Stats



MODELS USED

- Model 1 Small
- Model 1 Large
- Model 2
- Model 3



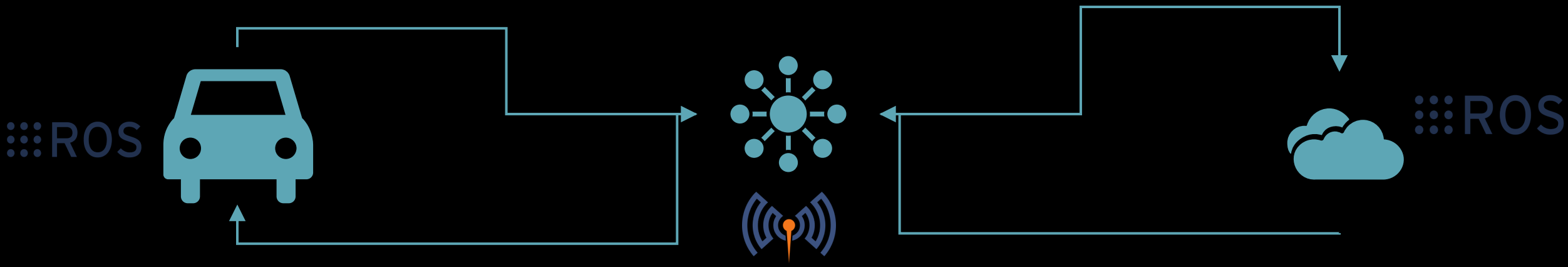
DEVICE USED

- CPU
- GPU

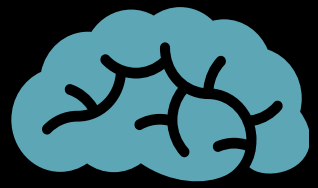
SYSTEM USED

- PC
- WS

MQTT-Based Serving Workflow

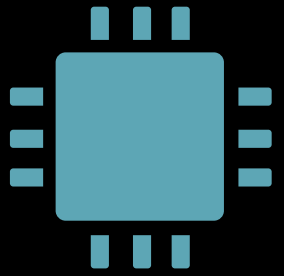


TFServing Stats



MODELS USED

- Model 2
- Model 3



DEVICE USED

- CPU
- GPU

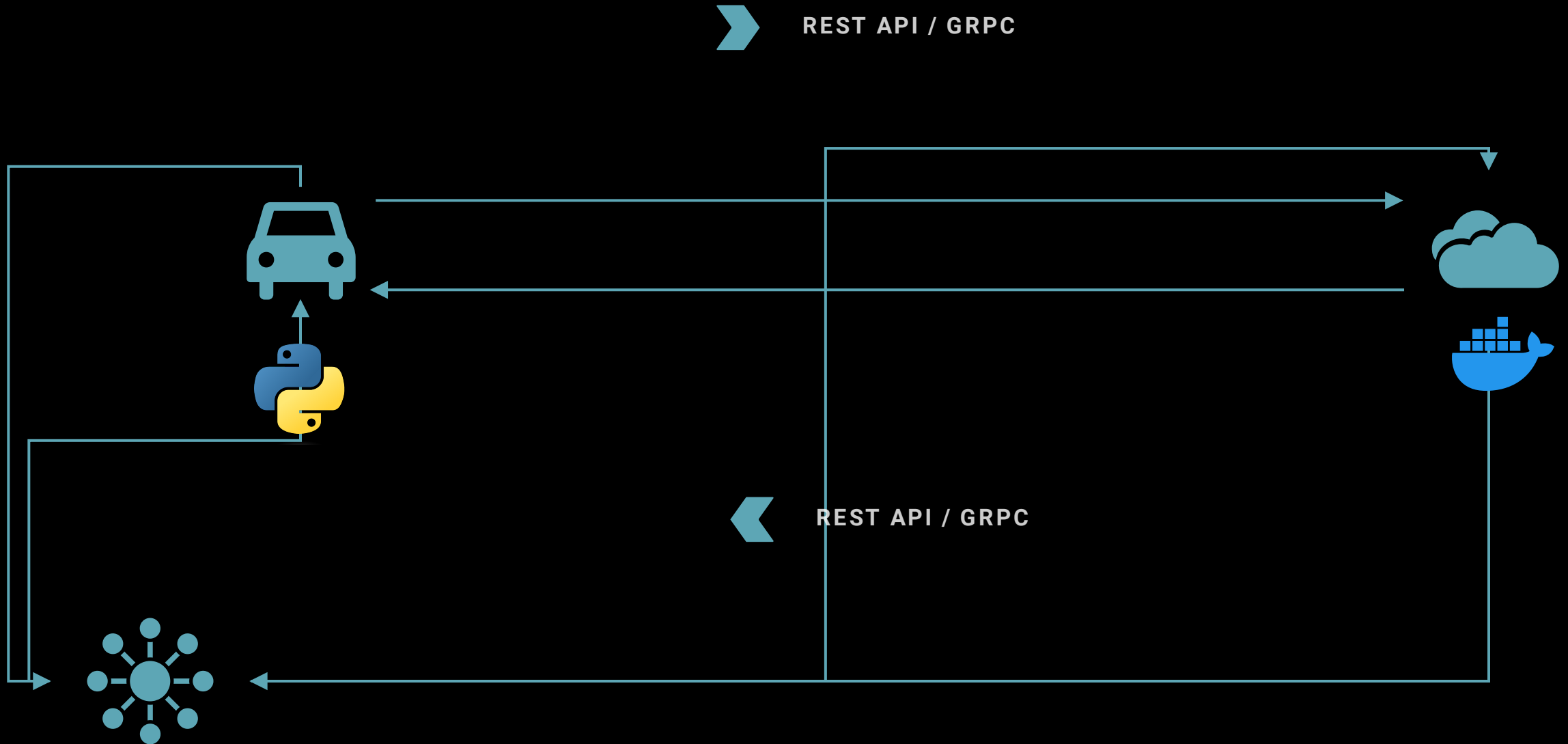
SYSTEM USED

- PC
- WS

COMMUNICATION PROTOCOL

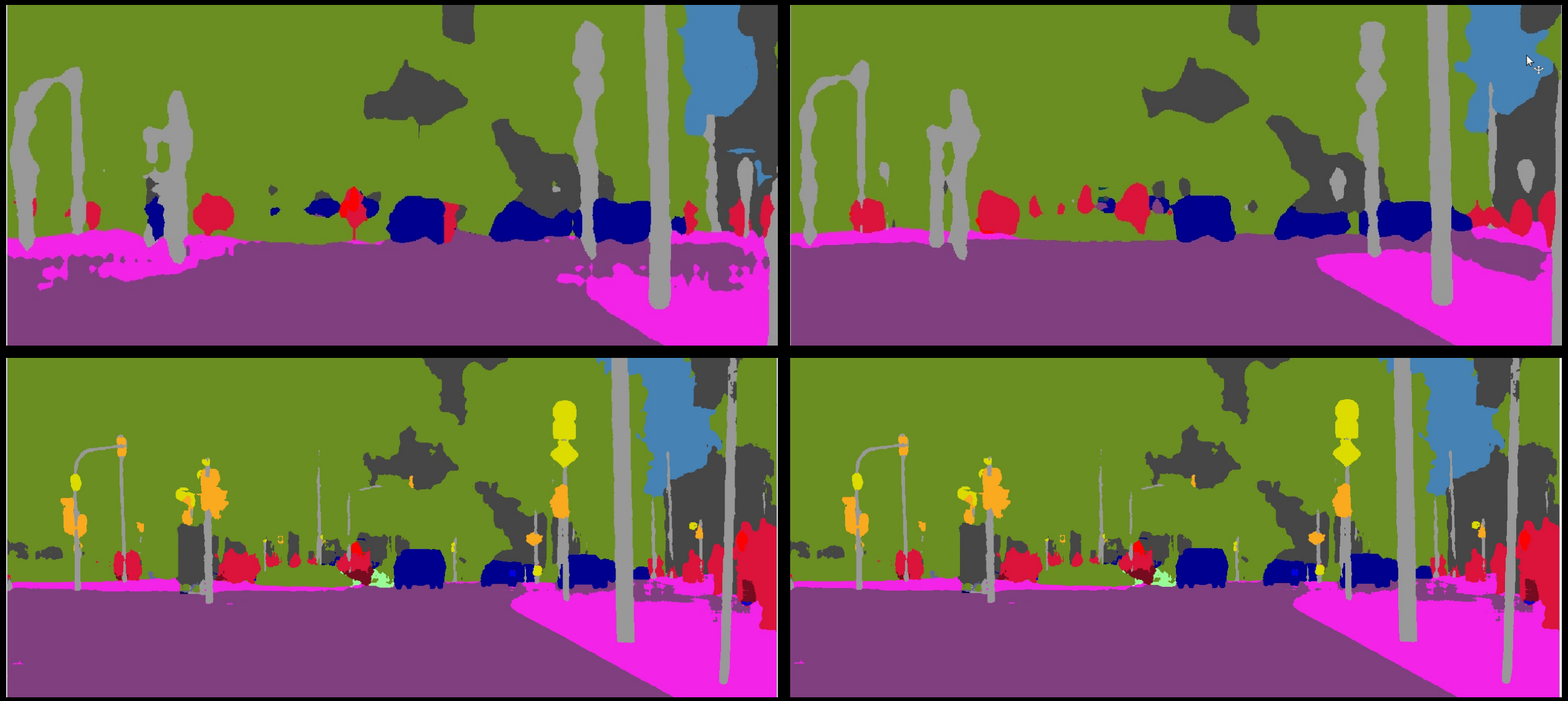
- HTTP/REST API
- gRPC

TFServing-Based Workflow



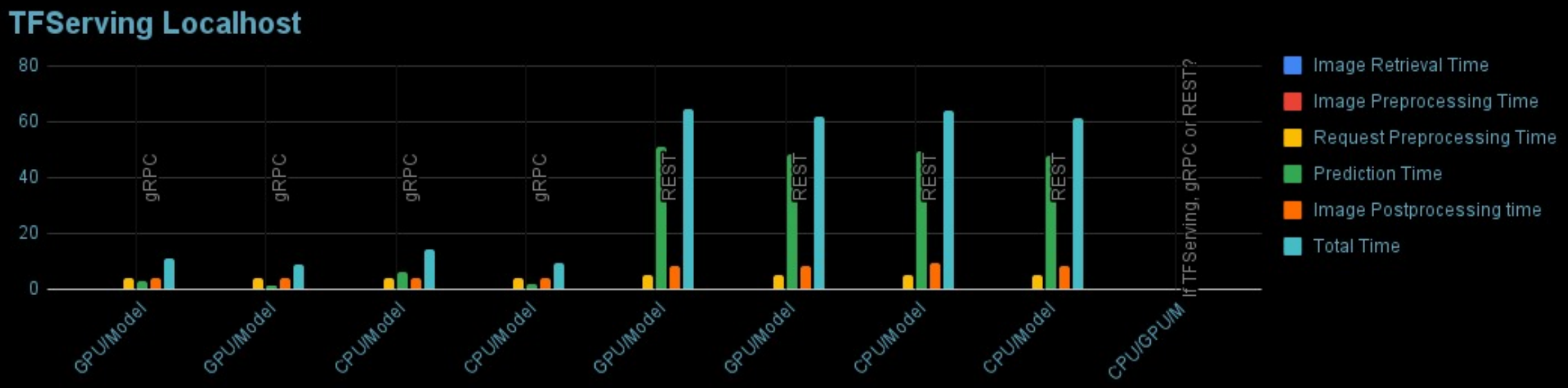
DEMO

RESULTS

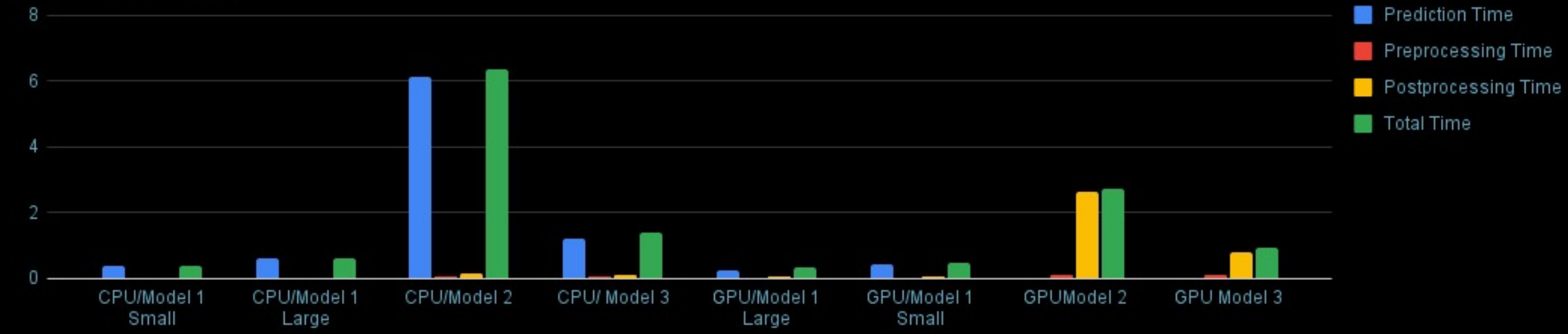


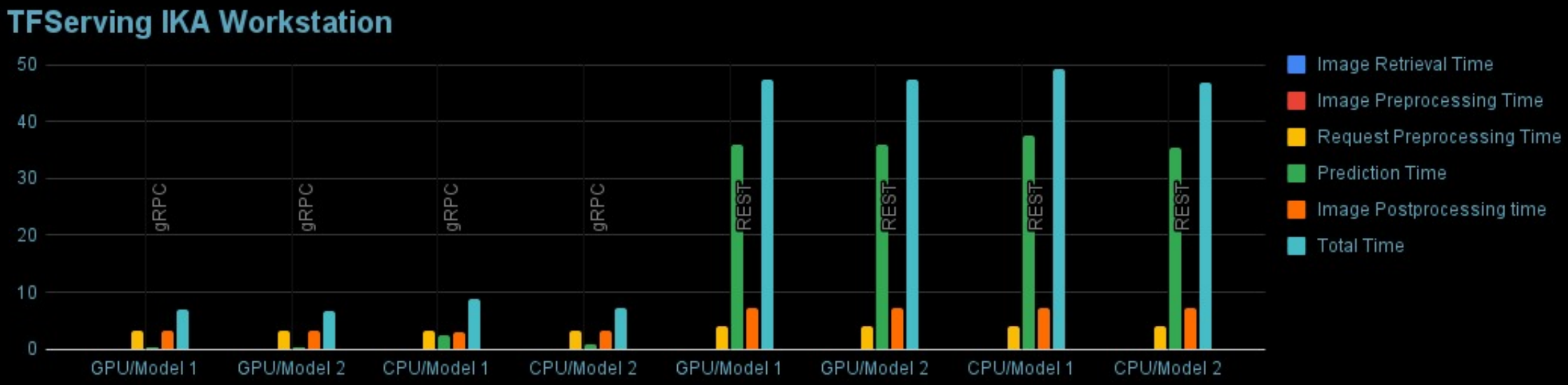
1,2 --> Frozen Graph outputs.

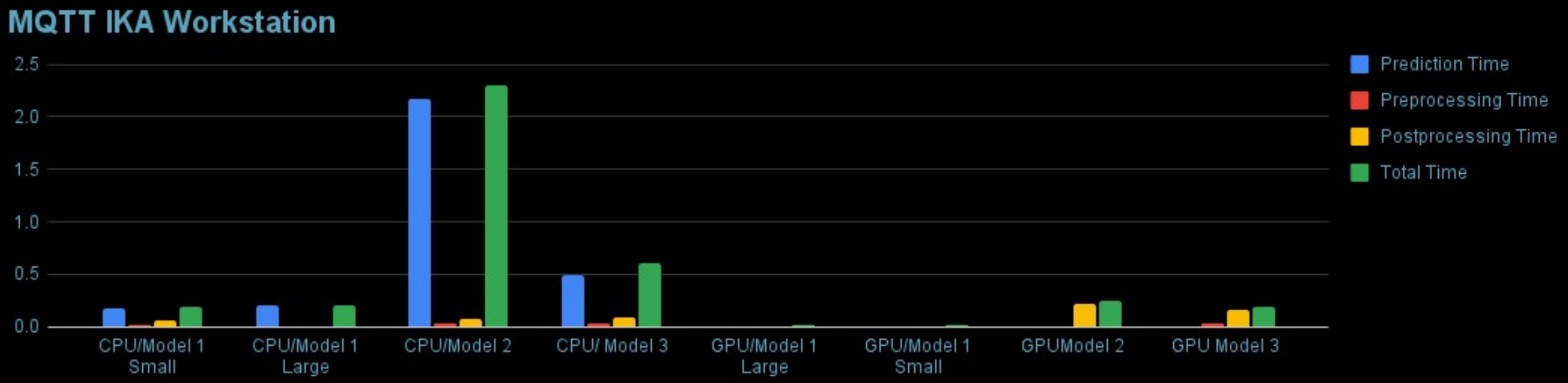
3,4 --> Model 2 and Model 3


TFServing Localhost
80
60
40
20
0
gRPC
gRPC
gRPC
gRPC
REST
REST
REST
REST
GPU/Model
GPU/Model
CPU/Model
CPU/Model
GPU/Model
GPU/Model
CPU/Model
CPU/Model
CPU/GPU/M
If TFServing, gRPC or REST?
Image Retrieval Time
Image Preprocessing Time
Request Preprocessing Time
Prediction Time
Image Postprocessing time
Total Time

MQTT Localhost




TFServing IKA Workstation
50
40
30
20
10
0
gRPC
gRPC
gRPC
gRPC
REST
REST
REST
REST
GPU/Model 1
GPU/Model 2
CPU/Model 1
CPU/Model 2
GPU/Model 1
GPU/Model 2
CPU/Model 1
CPU/Model 2
Image Retrieval Time
Image Preprocessing Time
Request Preprocessing Time
Prediction Time
Image Postprocessing time
Total Time


MQTT IKA Workstation
2.5
2.0
1.5
1.0
0.5
0.0
CPU/Model 1 Small
CPU/Model 1 Large
CPU/Model 2
CPU/ Model 3
GPU/Model 1 Large
GPU/Model 1 Small
GPUModel 2
GPU Model 3
Prediction Time
Preprocessing Time
Postprocessing Time
Total Time

Conclusion

- GRPC IS ABOUT 83% FASTER THAN REST BASED COMMUNICATION.
- PREDICTION TIME IS MUCH FASTER IN GPU THAN CPU
- POSTPROCESSING TAKES ALMOST >90% TIME WHEN IT COMES TO MQTT BASED SERVING WITH GPU AS COMPARED TO CPU.
- TFSERVING IS A BETTER APPROACH IN TERMS OF SETTING UP AND EASE OF USE AS COMPARED TO MQTT. HOWEVER MQTT IS FASTER IN TERMS OF INFERENCING DUE TO LACK OF REQUEST PROCESSING

Lessons Learnt

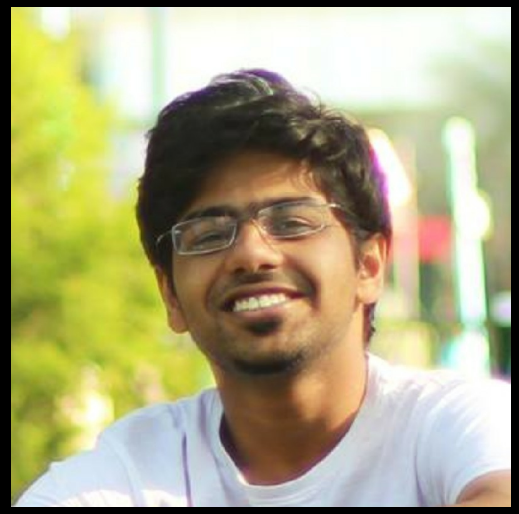
- EFFECTIVE USE OF GITLAB ISSUES, DOCKERS, PORT FORWARDING
- DEMOSAICING IMAGES FROM ROSBAG, ROSMSG_TO_CV, IMAGE POSTPROCESSING AND PREPROCESSING.
- WHY TFSERVING IS ACTUALLY 'BETTER' FOR ML PRODUCTION, REST VS GRPC
- FROZEN GRAPH VS SAVED MODELS

Future Work

- BENCHMARK TFSERVING AND MQTT-BASED SERVING ACROSS DIFFERENT SYSTEM, SIMULATING A VEHICLE-CLOUD INTERFACE
- TEST CONCURRENCY BY MAKING REQUESTS FROM MULTIPLE 'VEHICLE' NODE TO ONE CLOUD NODE (SMOOTH ON TFX, MQTT CAN'T HANDLE IT)
- TEST OUT NVIDIA TRITON SERVER

The Squad

AMOL KODANGE



ANUJOY CHAKRABORTY

PARTHAN MANISEKARAN

Thank you!