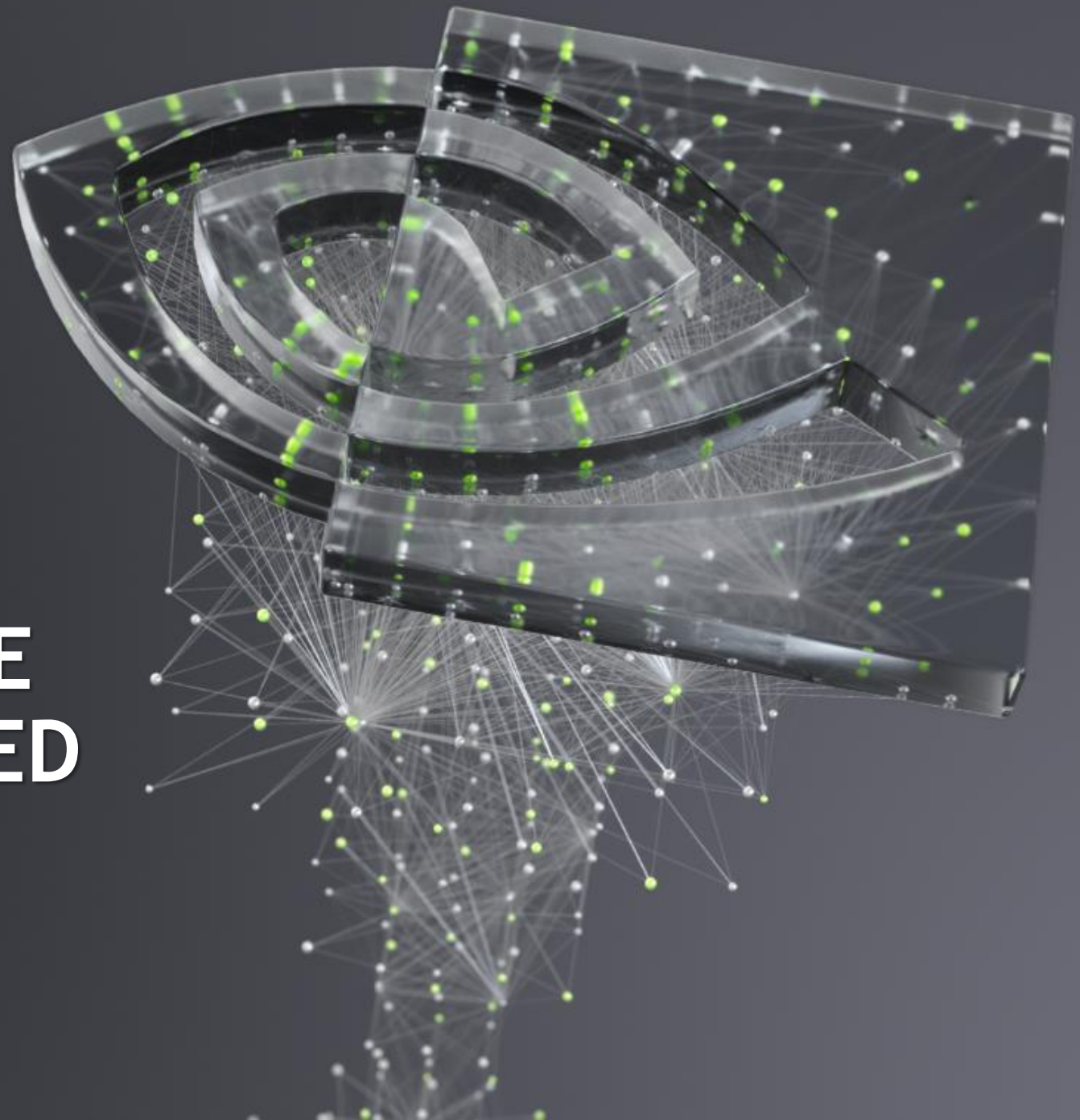


INFERENCE OF HUGE TRANSFORMER-BASED MODELS

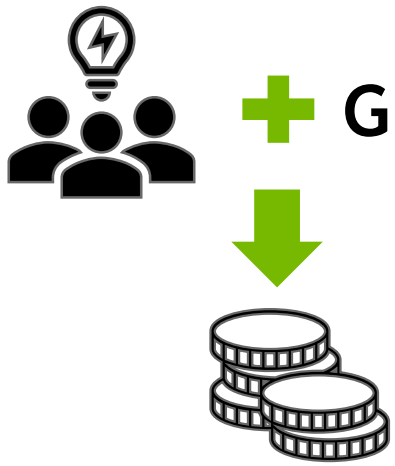
Denis Timonin, DL Solutions Architect

dtimonin@nvidia.com

3 MAIN STAGES FOR HUGE LANGUAGE MODEL USAGE

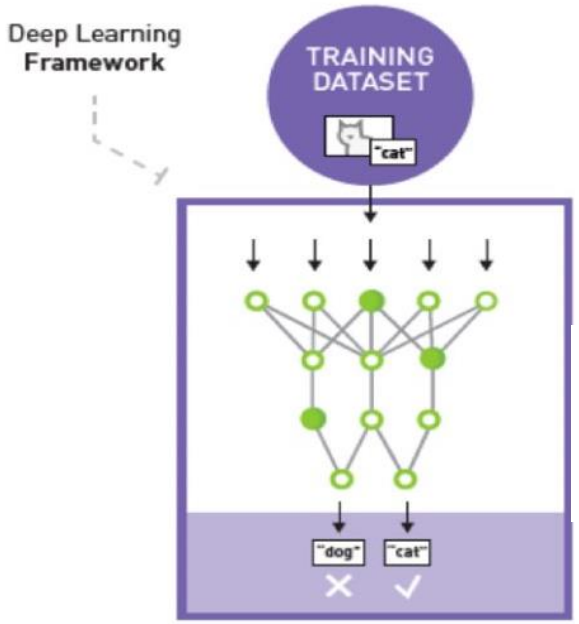
1 BUSINESS IDEA

Identifying a Business challenge that can be solved by GPT-3



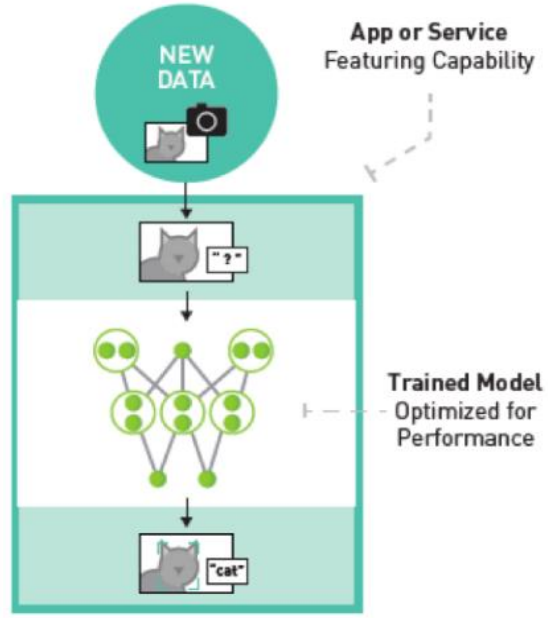
2 TRAINING

Learning a new capability from existing data



3 INFERENCE

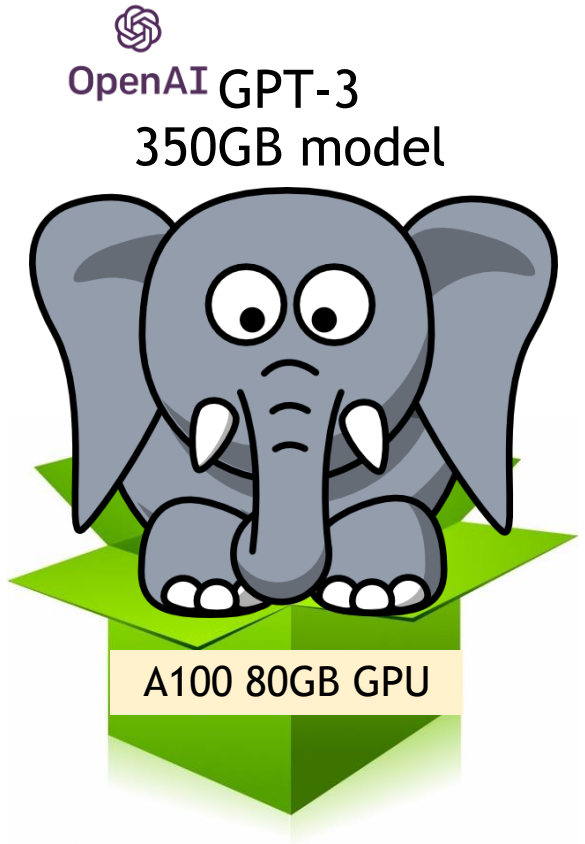
Applying this capability to new data



INFERENCE OF HUGE MODELS

Goals and Challenges

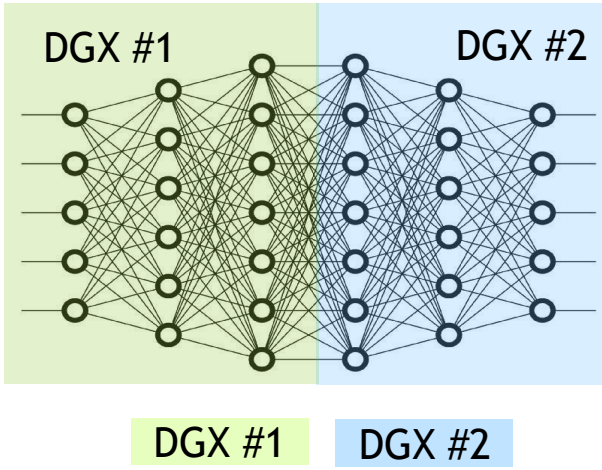
- **Goal: To infer huge models in an efficient and convenient way, including**
  - Maximizing Utilization of GPUs
  - A unified and simple inference solution for many models in production
  - Easier deployments, scaling and support
  - Maximizing Throughput, Minimizing Latency
- **Challenges:**
  - Huge model requires more memory than available on 1 GPU
  - There are no tools to infer Huge Models, apart from Triton
  - Model needs to be optimized/compile before the inference
  - Frameworks used for training Huge Models are quite complex and inadequate for inference

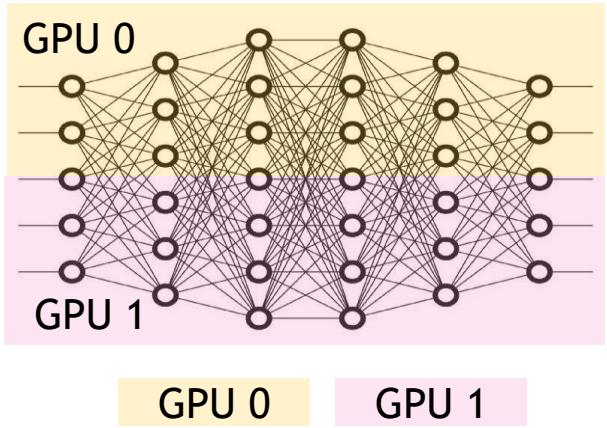


MEGATRON-LM MODEL PARALLELISM AND INFERENCE

Complementary Types of Model Parallelism

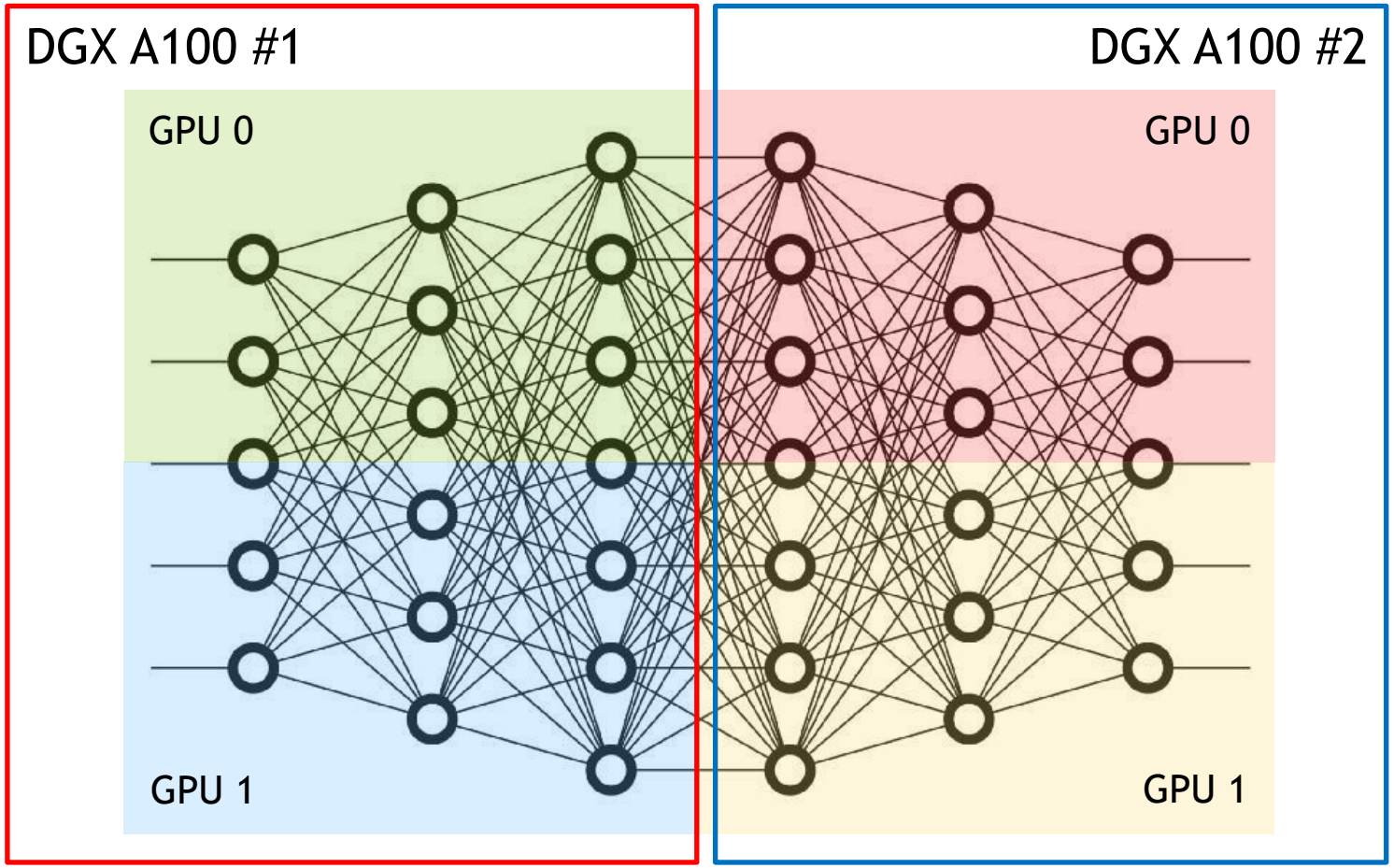
- Inter-Layer (Pipeline) Parallelism
  - Split sets of layers across multiple devices
  - ***Inference:***
    - ***Maximizes GPU utilization and Throughput***
    - ***Can be used easily with TRITON***
- Intra-Layer (Tensor) Parallelism
  - Split individual layers across multiple devices
  - ***Inference:***
    - ***Minimizes latency***



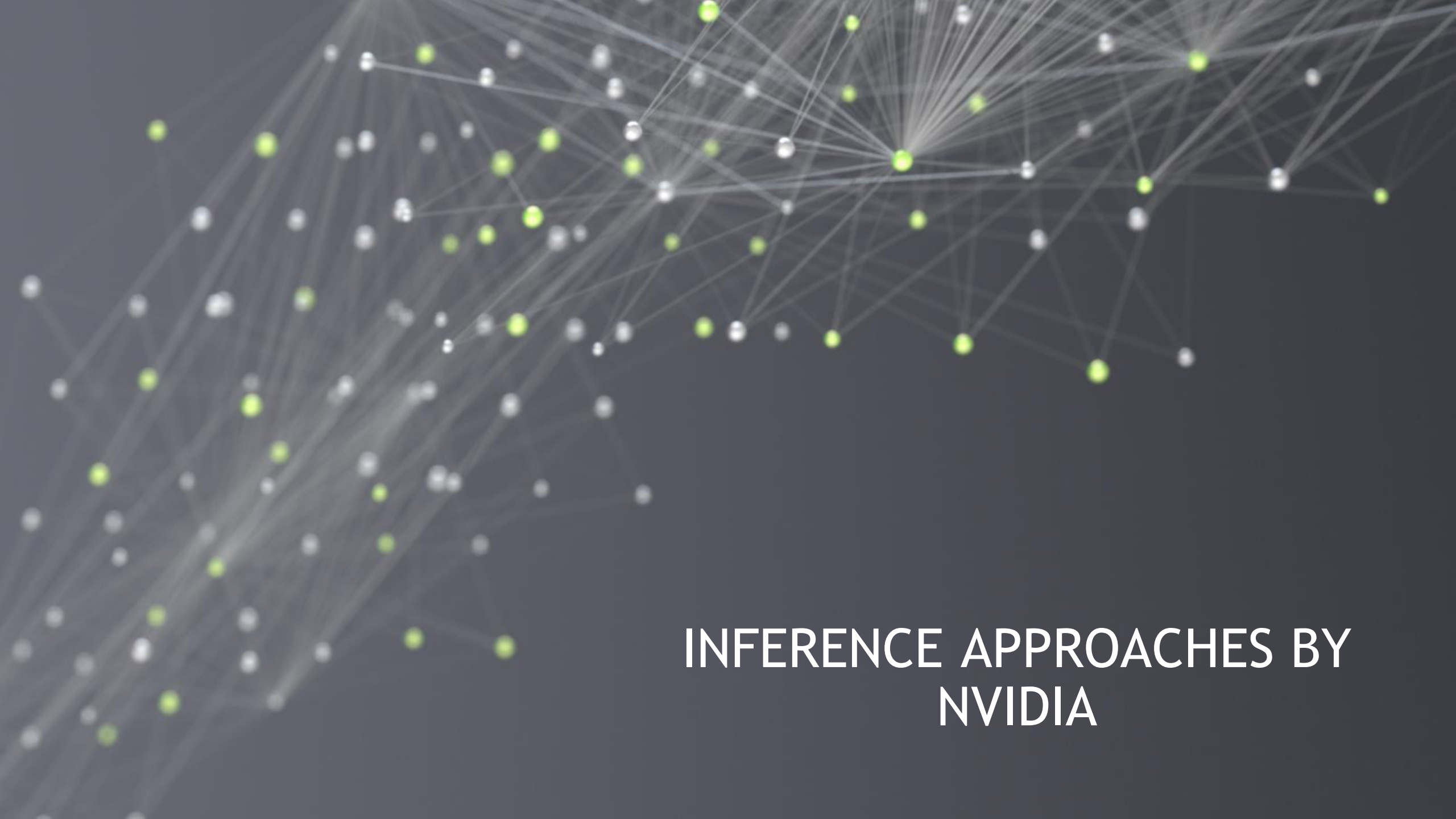


MEGATRON-LM MODEL PARALLELISM AND INFERENCE

Combined Model Parallelism. Multiple GPUs in Multiple DGXs.



Inter + Intra Parallelism



INFERENCE APPROACHES BY NVIDIA

TWO MAIN INFERENCE APPROACHES

Inference libraries by NVIDIA

FasterTransformer

https://github.com/NVIDIA/FasterTransformer
Special library created by NVIDIA for the inference of transformer-based models

Pros:
- Special for transformers and huge transformers
- Supports both tensor and pipeline parallelism techniques
- Fastest inference for GPT-3-like models
- Has Python bindings
- Integration with the TRITON inference server for fast deploy

Cons:
- Only strict types of models and layers are supported (BERT, GPT-2, Megatron-GPT-3). All other models like ViT-mixture won't work out of the box due lack support of Conv layers
- Complex to add support of new layers

TensorRT

https://developer.nvidia.com/tensorrt
Special compilation/optimization library created by NVIDIA for the inference wide range of NN models

Pros:
- Supports a lot of models and types of layers (may be good for ViT or layer-mixture models)
- Fastest inference BERT-like models
- Has Python bindings
- Integration with the TRITON inference server for fast deploy

Cons:
- No parallelism techniques out of the box
  Pipeline parallelism technique can be supported through TRITON
- Additional steps are needed to run Huge transformer model
- Quite slow for GPT-like autoregressive models


NVIDIA FASTER
TRANSFORMER

WHAT IS FASTER TRANSFORMER

Summary

Bo Yang Hsueh, NVIDIA, GTC 2020 : In-depth video about Faster Transformer
https://developer.nvidia.com/gtc/2020/video/s21417-vid

- FasterTransformer provides highly optimized transformer layer
  - Encoder transformer is based on BERT
  - Decoder transformer is based on **GPT-2, Megatron-GPT-3 and OpenNMT-tf**
  - Decoding contains whole process of translation, and it is also based on OpenNMT-tf
- Based on CUDA and cuBLAS
- Support both FP16 and FP32 **-> and INT8**
- Provide C++ API and TensorFlow Op
- Source codes are available ir https://github.com/NVIDIA/FasterTransformer

HOW TO USE IT

- You have to download and build FasterTrabsformer library
- Put the exported PyTorch weights of the pretrained Megatron-GPT-3, GPT-2, BERT, OpenNMT into project folder
- Run the script to split the weights onto partitions for multi-GPU tensor and pipeline parallelism
  - python ../sample/pytorch/utils/megatron_ckpt_convert.py \
    -i ./models/megatron-models/345m/release/ \
    -o ./models/megatron-models/c-model/345m/ -t_g 8 -i_g 1
- Run the script to start inference on multiple GPU-s
  - mpirun -n 8 --allow-run-as-root python ./pytorch/gpt_sample.py --tensor_para_size=8 --layer_para_size=1 \
    --ckpt_path="/workspace/fastertransformer/models/megatron-models/c-model/345m/8-gpu”
- Demo and instruction for GPT-3 https://github.com/NVIDIA/FasterTransformer#gpt-demo

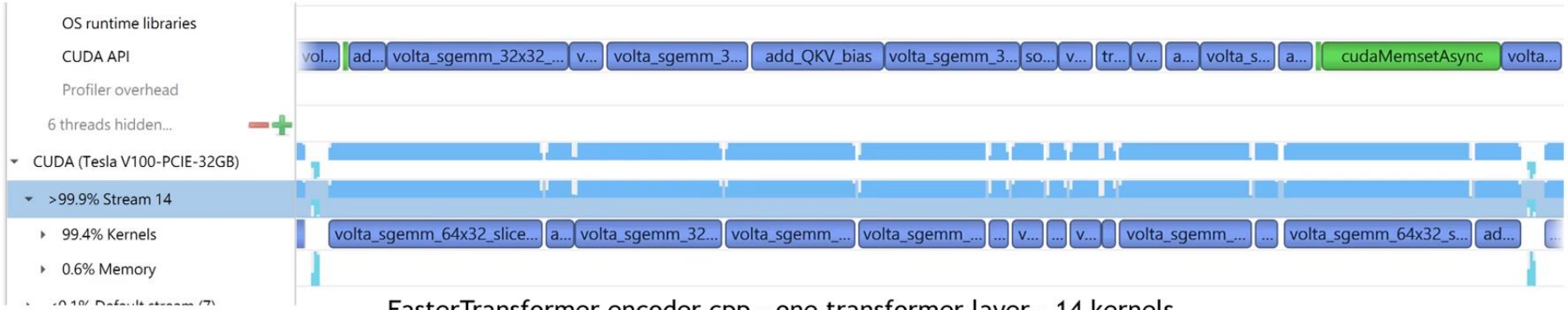
- Inference acceleration from x1.5-x4 for Megatron GPT-3 model

HOW IT WORKS

Encoder Inference in the Framework:

TensorFlow encoder no XLA - one transformer layer - 50 kernels
batch size 1, 12 heads, size per head 64, FP32

Encoder Inference in the FasterTransformer:



FasterTransformer encoder cpp - one transformer layer - 14 kernels
batch size 1, 12 heads, size per head 64, FP32

Bo Yang Hsueh, NVIDIA, GTC 2020 : In-depth video about Faster Transformer

https://developer.nvidia.com/gtc/2020/video/s21417-vid

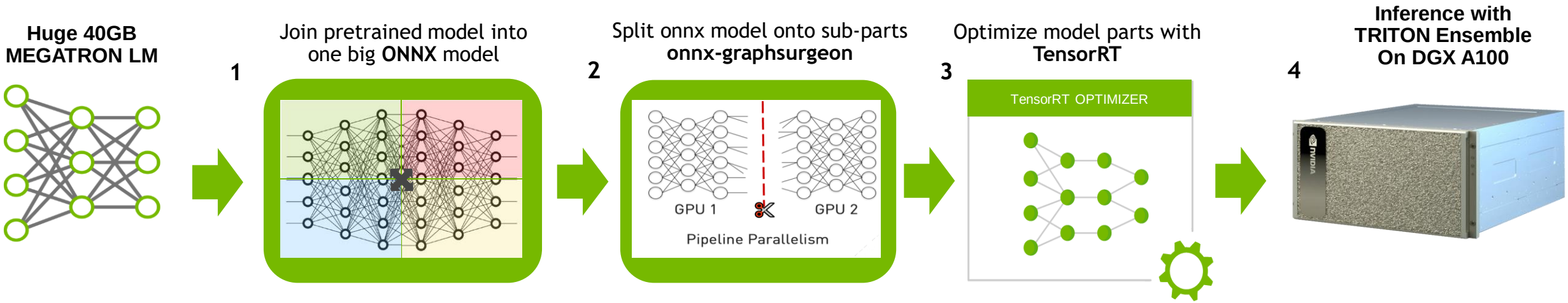
TENSORT +ONNX+TRITON

MEGATRON-LM GPT-3

Pipeline-parallelism Inference steps

Denis Timonin, GTC 2021:
Megatron GPT-3 Large Model Inference with Triton and ONNX Runtime
https://www.nvidia.com/en-us/on-demand/session/gtcspring21-s31578/

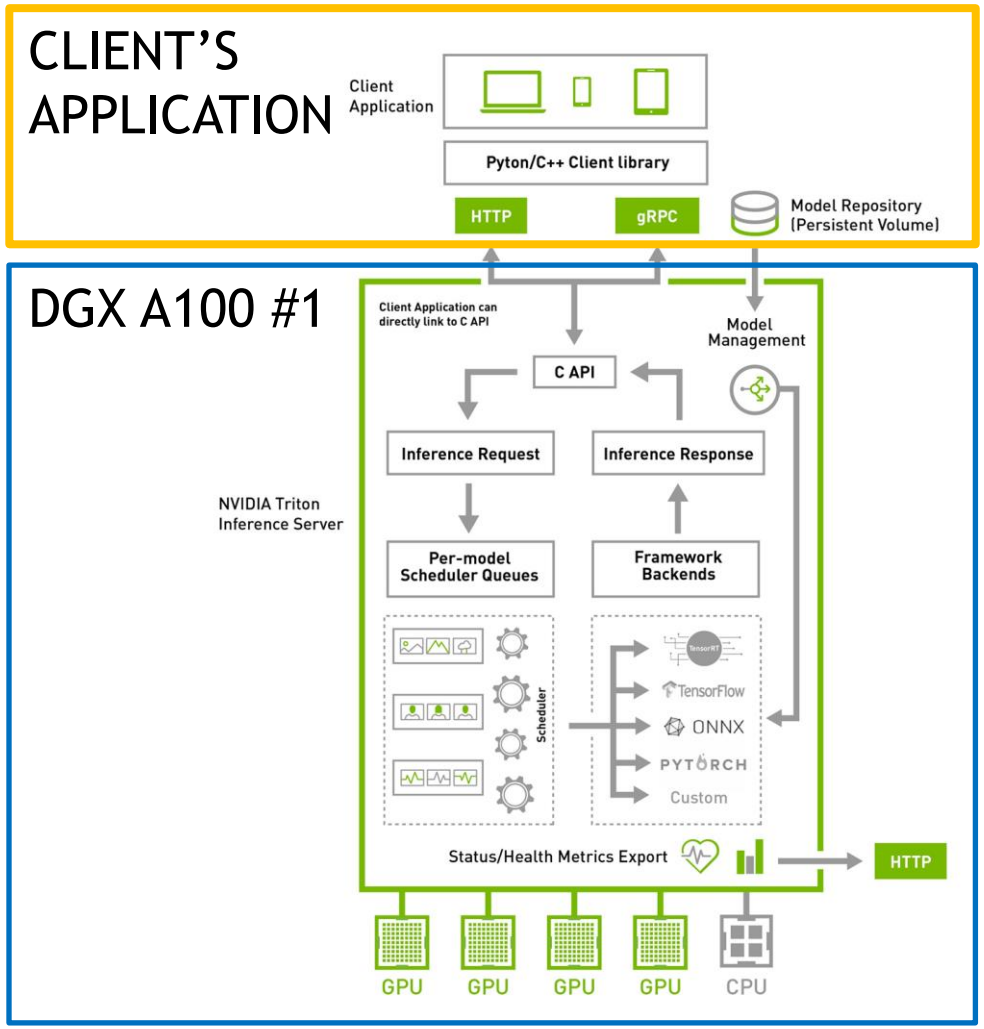
Steps to run our huge model in inference using pipeline-parallelism technique:



- Inference acceleration from x1.5-x4 for Megatron BERT model

TRITON: INFERENCE SERVER ARCHITECTURE

Easy to Use



Pretrained Neural Network is placed on DGX and ready for inference with TRITON

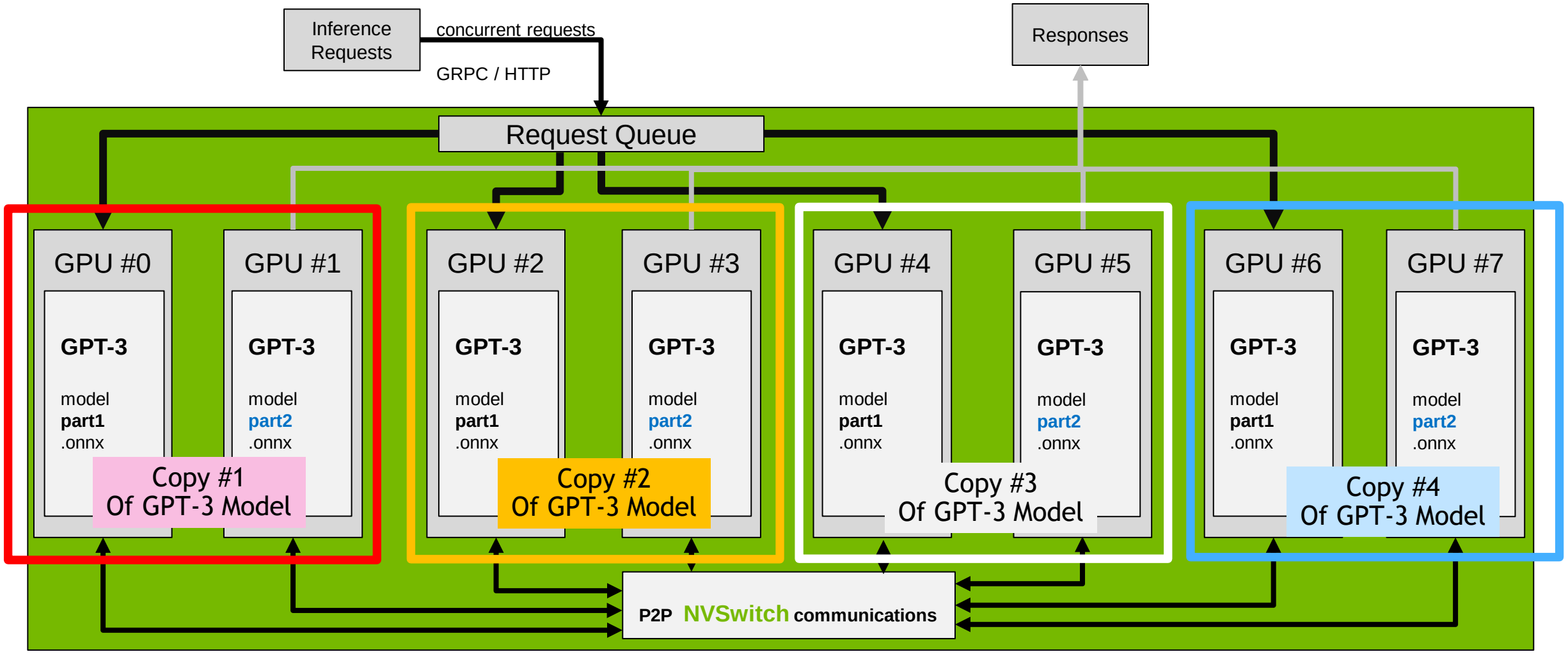
https://github.com/triton-inference-server/server

TRITON's Ensembling technique is needed to run model in the **pipeline-parallelism mode**

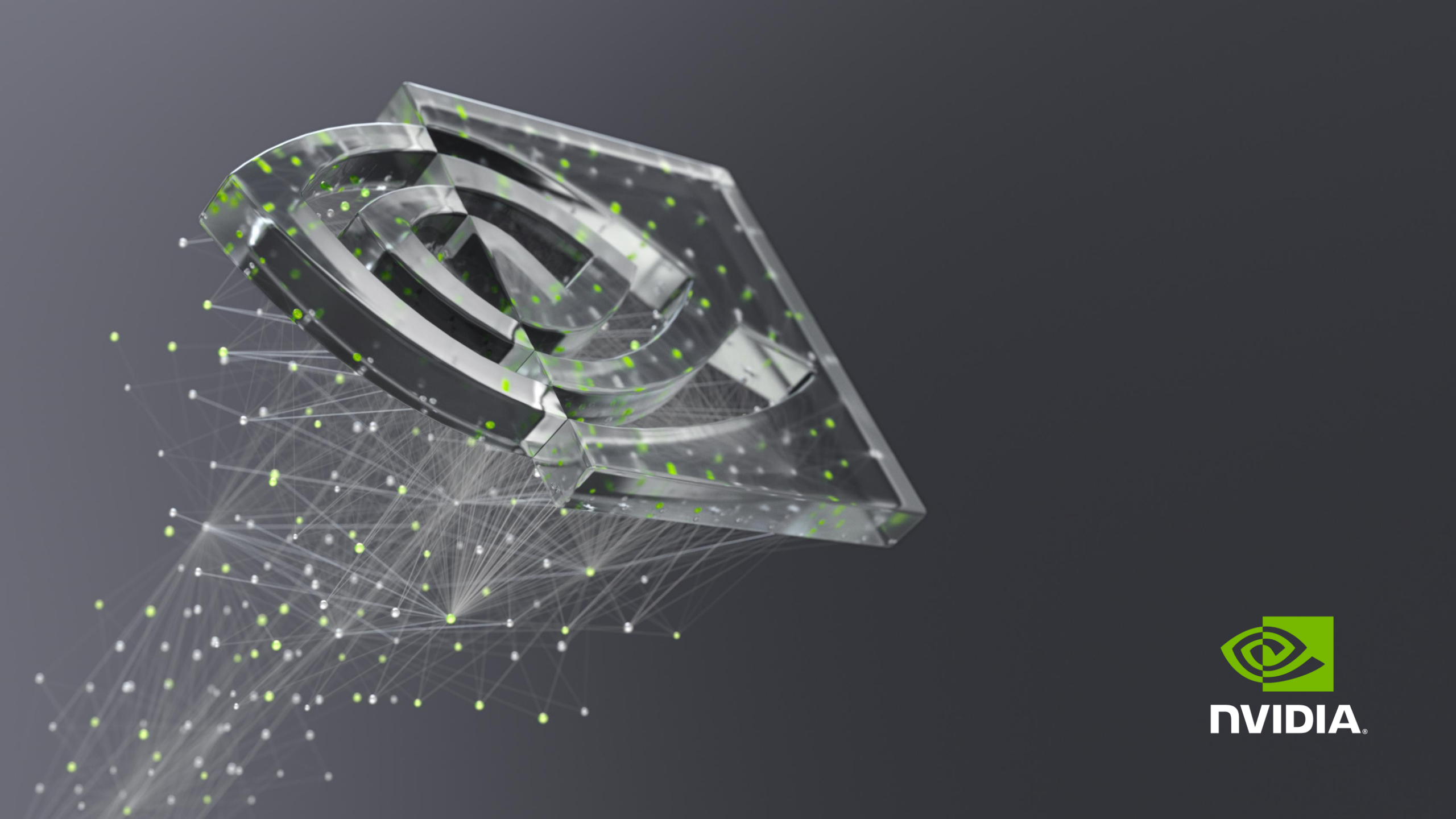
https://github.com/triton-inference-server/server/blob/main/docs/architecture.md#ensemble-models

SCALING BY ADDING ONE SIMPLE LINE OF CODE

Running 4 Different Inference Jobs on one DGX A100



Inference on DGX A100 with 4 x 2 x GPUs used for our model


NVIDIA