NVIDIA GTC

CONNECT WITH THE EXPERTS

CUDA, TensorRT & DriveWorks on DRIVE AGX

(CWE 21185)

Aaraadhya Narra, Product Manager, Robotaxis, Autonomous Vehicles
Anurag Dixit, Deep Learning Software Engineer
Dennis Lui, Manager, Solutions Architect
Josh Park, Automotive Devtech Manager
Yu-Te Cheng, Senior Deep Learning Software Engineer

NVIDIA GTC

Frequently Asked Questions

(CWE21185)

Q. How do I create an engine that is optimized for several different batch sizes?

To optimize for multiple different batch sizes, create optimization profiles at the dimensions that are assigned to OptProfilerSelector::kOPT

Q. Are engines and calibration tables portable across TensorRT versions?

No, engines and calibration tables are not guaranteed to be binary compatible with different versions of TensorRT

NVIDIA GTC

Frequently Asked Questions

(CWE21185)

Q. How do I use TensorRT on multiple GPUs?

Each ICudaEngine object is bound to a specific GPU when it is instantiated, either by the builder or on deserialization. To select the GPU, use cudaSetDevice() before calling the builder or deserializing the engine. Each IExecutionContext is bound to the same GPU as the engine from which it was created. When calling execute() or enqueue(), ensure that the thread is associated with the correct device by calling cudaSetDevice() if necessary.

Q. How do I get the version of TensorRT from the library file?

$ nm -D libnvinfer.so.4.1.0 | grep tensorrt_version

000000000c18f78c B tensorrt_version_4_0_0_7

NVIDIA GTC

Frequently Asked Questions

(CWE21185)

Q. What can I do if my network is producing the wrong answer?

Several possible reasons. Some of the troubleshooting methods:

- Turn on INFO level messages from the log stream and check what TensorRT is reporting.
- Check that your input preprocessing is generating exactly the input format required by the network.
- If you're using reduced precision, run the network in FP32. If it produces the correct result, it is possible that lower precision has an insufficient dynamic range for the network.
- Try marking intermediate tensors in the network as outputs and see if they match what you are expecting. Note: Marking tensors as outputs may inhibit optimizations, and therefore, may change the results.

NVIDIA GTC

Frequently Asked Questions

(CWE21185)

Q. How do I choose the optimal workspace size?

IBuilderConfig::setMaxWorkspaceSize() controls the maximum amount of workspace that may be allocated and will prevent algorithms that require more workspace from being considered by the builder.

At runtime, the space is allocated automatically when creating an IExecutionContext. The amount allocated will be no more than is required, even if the amount set in IBuilderConfig::setMaxWorkspaceSize() is much higher. Applications should therefore allow the TensorRT builder as much workspace as they can afford; at runtime TensorRT will allocate no more than this, and typically less.

Q. Is INT4 quantization or INT16 quantization supported by TensorRT?

Neither INT4 nor INT16 quantization are supported by TensorRT at this time.

NVIDIA GTC

Frequently Asked Questions

(CWE21185)

Q. If I build the engine on one GPU and run the engine on another GPU, will this work?

We recommend that you don't, however, if you do, you'll need to follow these guidelines:

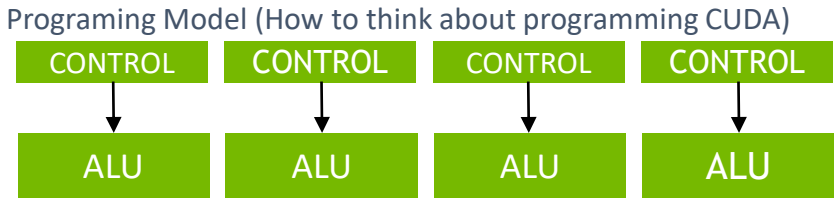
- Major, minor and patch version of TensorRT must match between systems.
- The CUDA compute capability major and minor versions must match between systems.
- The following properties should match between systems:
    - Maximum GPU graphics clock speed
    - Maximum GPU memory clock speed
    - GPU memory bus width
    - Total GPU memory
    - GPU L2 cache size
    - SM processor count
    - Asynchronous engine count

    If any of the above properties do not match, you will receive the following warning: *Using an engine plan file across different models of devices is not recommended and is likely to affect performance or even cause errors.*

    If you still want to proceed, then you should build the engine on the smallest SKU in the family because autotuner choices made on smaller GPUs will generalize better.

CUDA

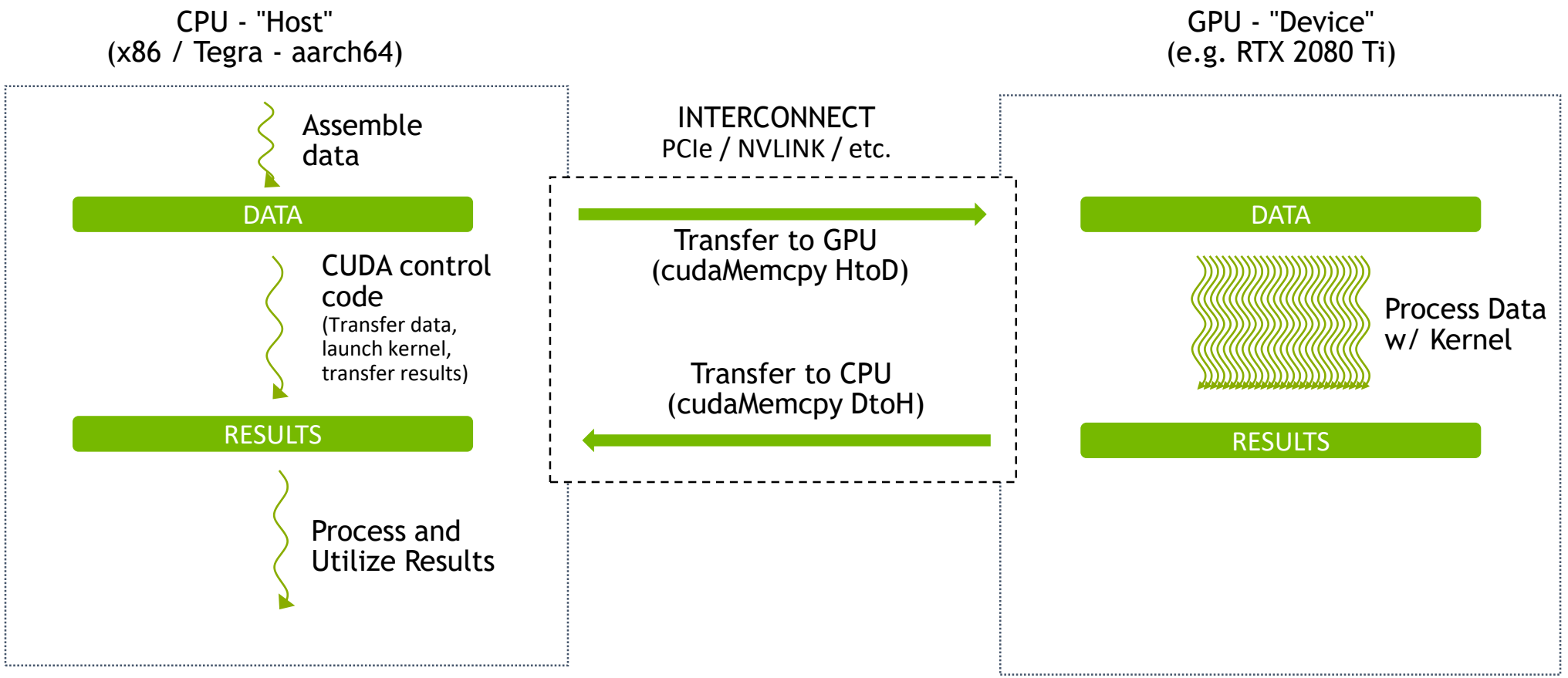
- CUDA is a parallel computing platform and programming model from NVIDIA for General Purpose GPU (GPGPU) programming.
- GPU tasks are programmed as CUDA kernels executed in SIMT (Single Instruction Multiple Threads) programming paradigm (More later on considerations when writing CUDA)



- What CUDA handles?
  - Thread organization
  - Memory management on GPU/CPU
  - Ways to synchronize across multiple threads to accomplish a common task.

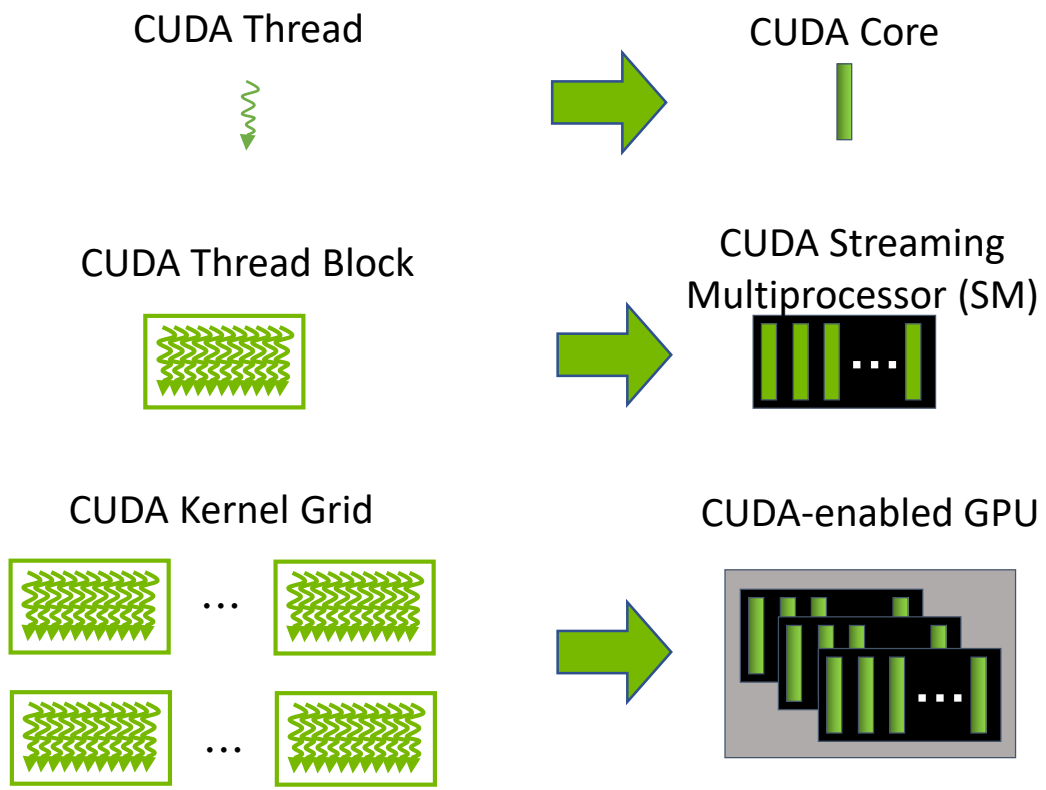
NVIDIA GTC

GPGPU Computing – Workflow



CUDA Kernel Execution

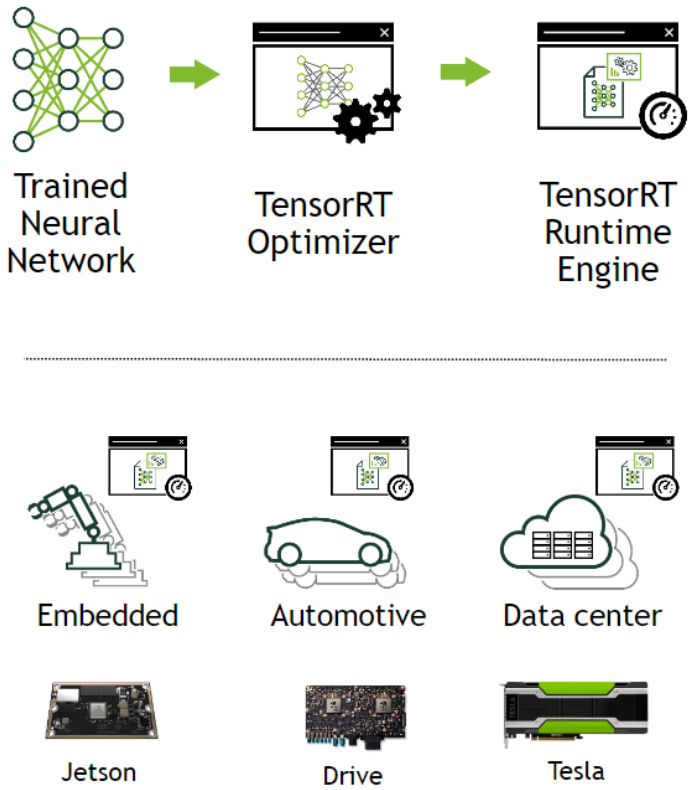
- Each thread is executed by a core
- 32 threads are grouped together to form a warp to run in a lock-step manner (SIMT)
- Thread block is group of threads that can span multiple warps
- Each block is executed by one SM and does not migrate
- Several concurrent blocks can reside on one SM depending resource availability / requirements
- Each kernel is executed on one device



NVIDIA GTC

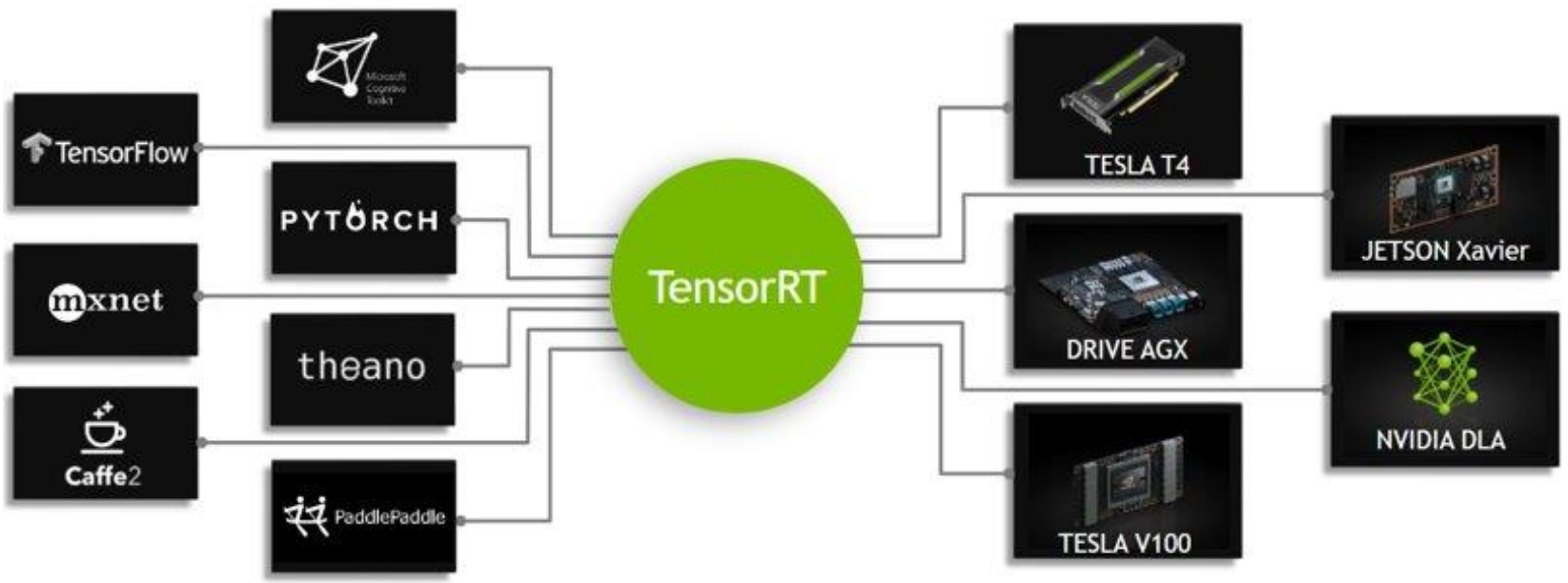
TensorRT

- A platform for high-performance deep learning inference
- Optimize and deploy neural networks in production environments
- Maximize throughput for latency-critical apps
- Memory-efficient apps with reduced precisions (FP16 and INT8)
- Run multiple models on a node with containerized inference server



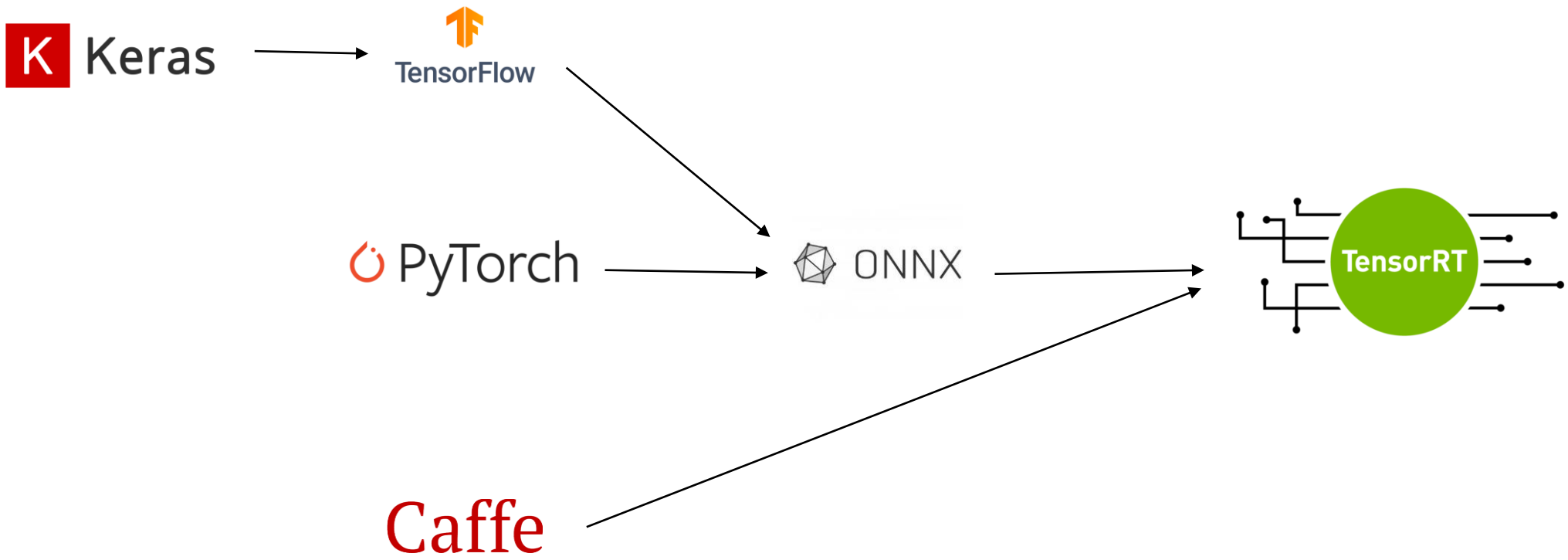
NVIDIA GTC

TensorRT Overview



NVIDIA GTC

How to import models to TensorRT

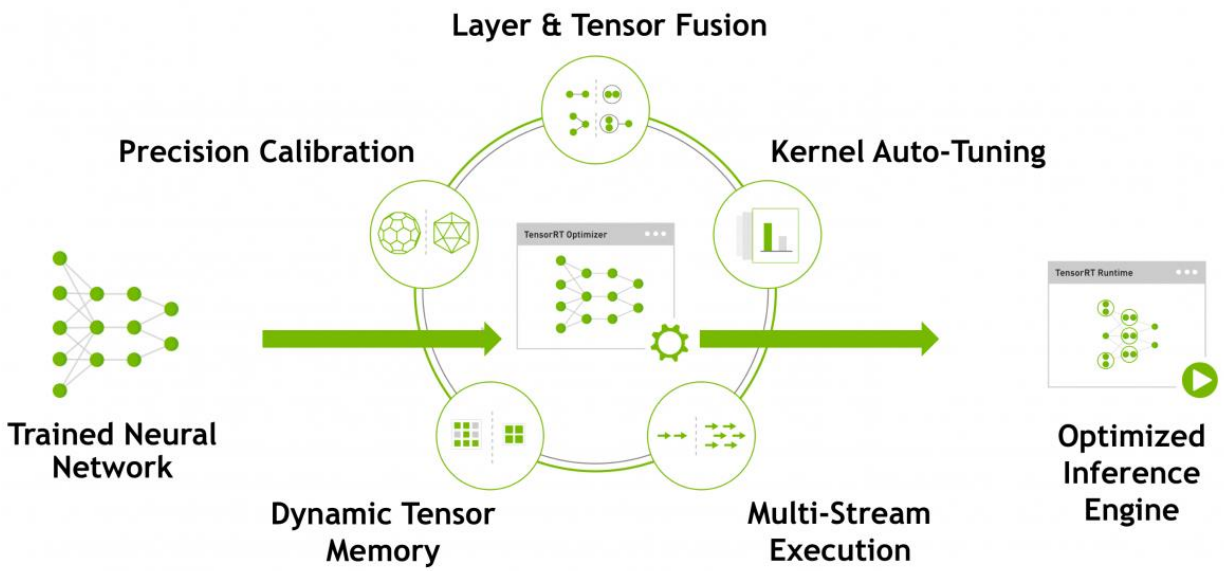


Note: *Other frameworks have direct conversion or go through ONNX format*

NVIDIA GTC

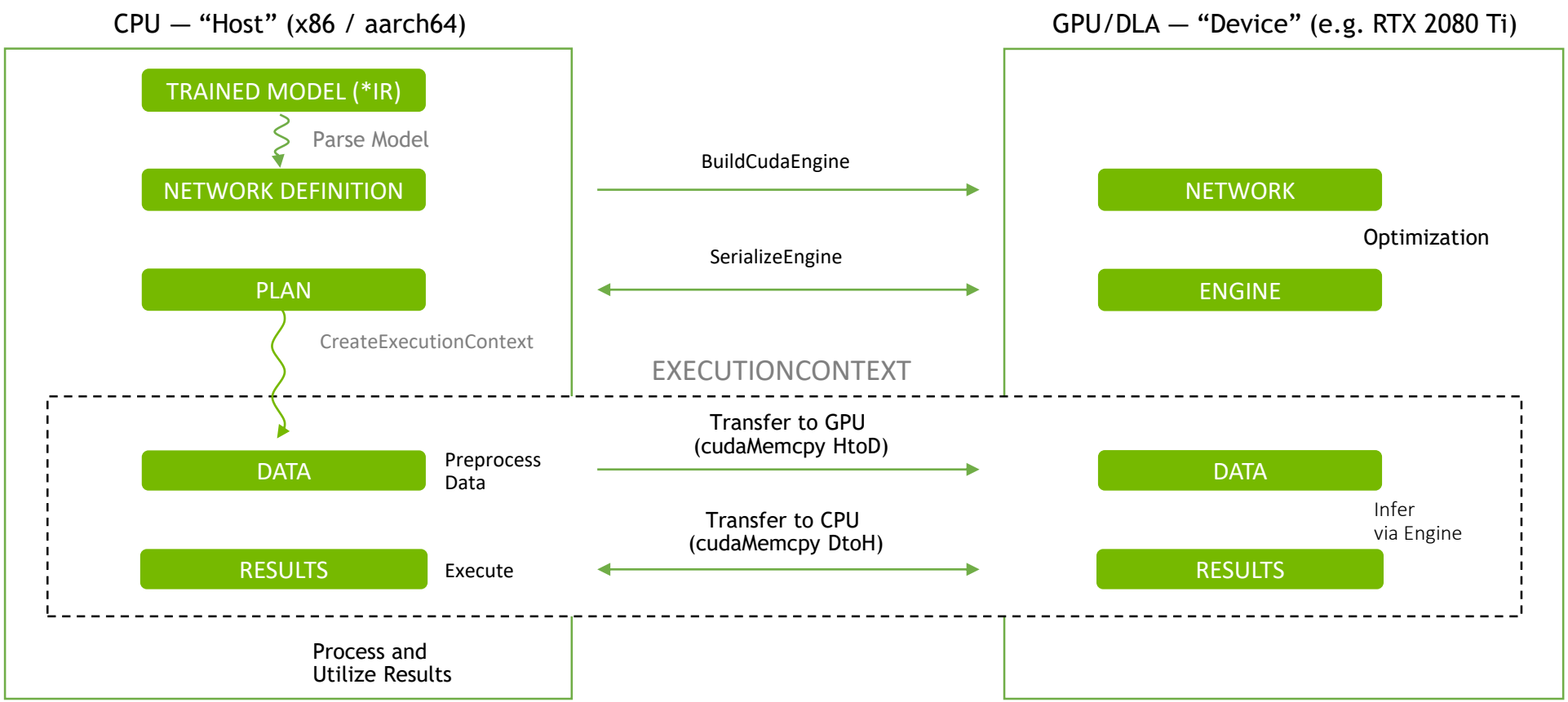
TensorRT Workflow

- Layer & Tensor Fusion
- Auto-Tuning
- Precision Calibration
- Multi-Stream Execution
- Dynamic Tensor Memory



NVIDIA GTC

TensorRT Application Workflow



** IR: Intermediate Representation (e.g. ONNX)*