Data Science

English

Real-time Serving for XGBoost, Scikit-Learn RandomForest, LightGBM, and More

Feb 02, 2022

+4 Like Discuss (0)

By William Hicks

efforts to accelerate the deployment of tree-based models (including random forest and gradient-boosted models) have received less attention, despite their continued dominance in tabular data analysis and their importance for use-cases where interpretability is essential.

As organizations like DoorDash and Capital One turn to tree-based models for the analysis of massive volumes of mission-critical data, it has become increasingly important to provide tools to help make deploying such models easy, efficient, and performant.

NVIDIA Triton Inference Server offers a complete solution for deploying deep learning models on both CPUs and GPUs with support for a wide variety of frameworks and model execution backends, including PyTorch, TensorFlow, ONNX, TensorRT, and more. Starting in version 21.06.1, to complement NVIDIA Triton Inference Server existing deep learning capabilities, the new Forest Inference Library (FIL) backend provides support for tree models, such as XGBoost, LightGBM, Scikit-Learn Random Forest, RAPIDS cuML Random Forest, and any other model supported by Treelite.

Based on the RAPIDS Forest Inference Library (FIL), the NVIDIA Triton Inference Server FIL backend allows users to take advantage of the same features of the NVIDIA Triton Inference Server they use to achieve optimal throughput/latency for deep learning models to deploy tree-based models on the same system.

In this post, we'll provide a brief overview of the NVIDIA Triton Inference Server itself then dive into an example of how to deploy an XGBoost model using the FIL backend. Using NVIDIA GPUs, we will see that we do not always have to choose between deploying a more accurate model or keeping latency manageable.

In the example notebook, by taking advantage of the FIL backend's GPU-accelerated inference on an NVIDIA DGX-1 server with eight V100 GPUs, we'll be able to deploy a much more sophisticated fraud detection model than we would be able to on CPU while keeping p99 latency under 2ms and *still* offer over 400K inferences per second (630 MB/s) or about 20x higher throughput than on CPU.

NVIDIA Triton Inference Server

Join the **NVIDIA Triton and NVIDIA TensorRT community** to stay current on the latest product updates, bug fixes, content, best practices, and more.

machine learning models. Designed to make the process of performant model deployment as simple as possible, NVIDIA Triton Inference Server provides solutions to many of the most common problems encountered when attempting to deploy ML algorithms in real-world applications, including:

- *Multi-Framework Support***:** Supports all of the most common deep learning frameworks and serialization formats, including PyTorch, TensorFlow, ONNX, TensorRT, OpenVINO, and more. With the introduction of the FIL backend, NVIDIA Triton Inference Server also provides support for XGBoost, LightGBM, Scikit-Learn/cuML RandomForest, and Treelite-serialized models from any framework.
- *Dynamic Batching***:** Allows users to specify a batching window and collate any requests received in that window into a larger batch for optimized throughput.
- *Multiple Query Types*: Optimizes inference for multiple query types: real time, batch, streaming, and also supports model ensembles.
- *Pipelines and Ensembles***:** Models deployed with NVIDIA Triton Inference Server can be connected in sophisticated pipelines or ensembles to avoid unnecessary data transfers between client and server or even host and device.
- *CPU Model Execution***:** While most users will want to take advantage of the substantial performance gains offered by GPU execution, NVIDIA Triton Inference Server allows you to run models on either CPU or GPU to meet your specific deployment needs and resource availability.
- *Customization***:** If NVIDIA Triton Inference Server does not provide support for part of your pipeline, or if you need specialized logic to link together various models, you can add precisely the logic you need with a custom Python or C++ backend.
- *Run anywhere*: On scaled-out cloud or data center, enterprise edge, and even on embedded devices. It supports both bare metal and virtualized environments (e.g. VMware vSphere) for AI inference.
- *Kubernetes and AI platform support*:
  - Available as a Docker container and integrates easily with Kubernetes platforms like AWS EKS, Google GKE, Azure AKS, Alibaba ACK, Tencent TKE or Red Hat OpenShift.
  - Available in Managed CloudAI workflow platforms like Amazon SageMaker, Azure ML, Google Vertex AI, Alibaba Platform for AI Elastic Algorithm Service, and Tencent TI-EMS.
- *Enterprise support*: NVIDIA AI Enterprise software suite includes full support of NVIDIA Triton Inference Server, such as access to NVIDIA AI experts for deployment and management guidance, prioritized notification of security fixes and maintenance releases, long term support (LTS) options and a designated support agent.

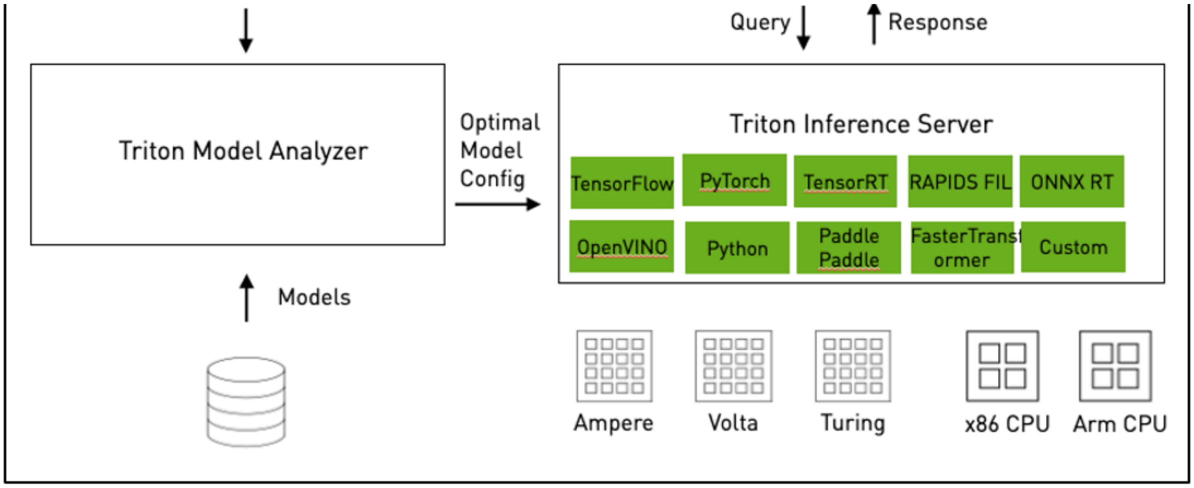


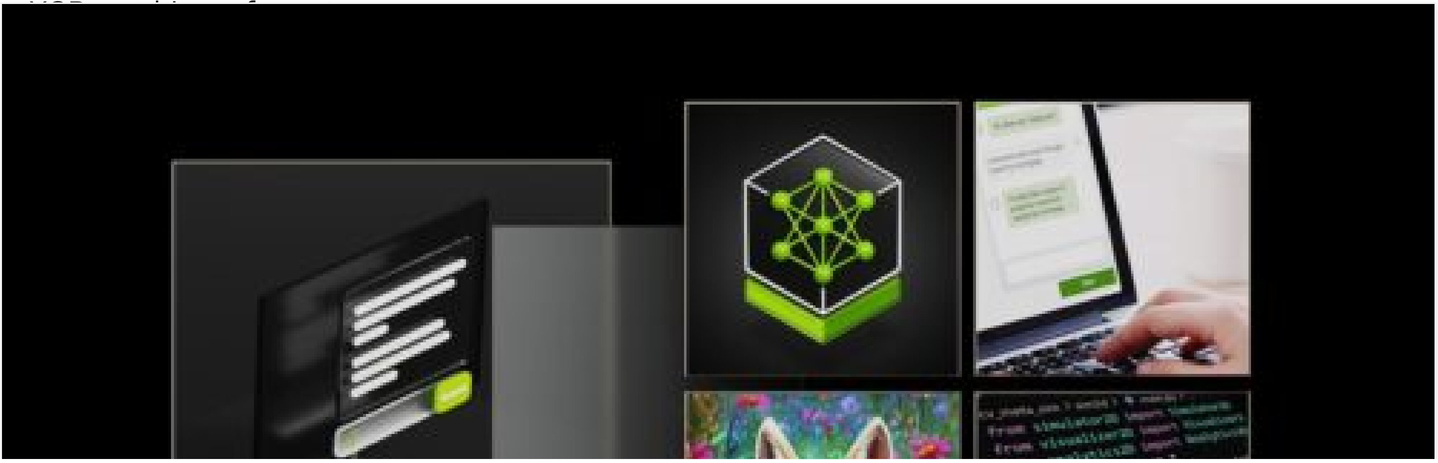
Figure 1: NVIDIA Triton Inference Server Architecture Diagram.

To get a better sense of how we can take advantage of some of these features with the FIL backend for deploying tree models, let's look at a specific use case.

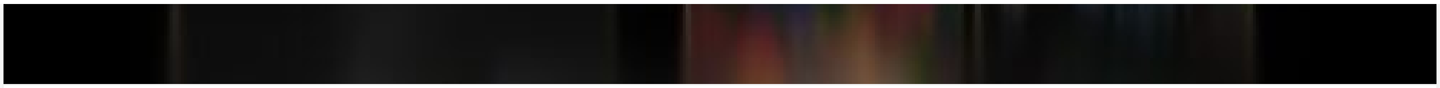
Example: Fraud Detection with the FIL Backend

In order to deploy a model in NVIDIA Triton Inference Server, we need a configuration file specifying some details about deployment options and the serialized model itself. Models can currently be serialized in any of the following formats:

Related posts

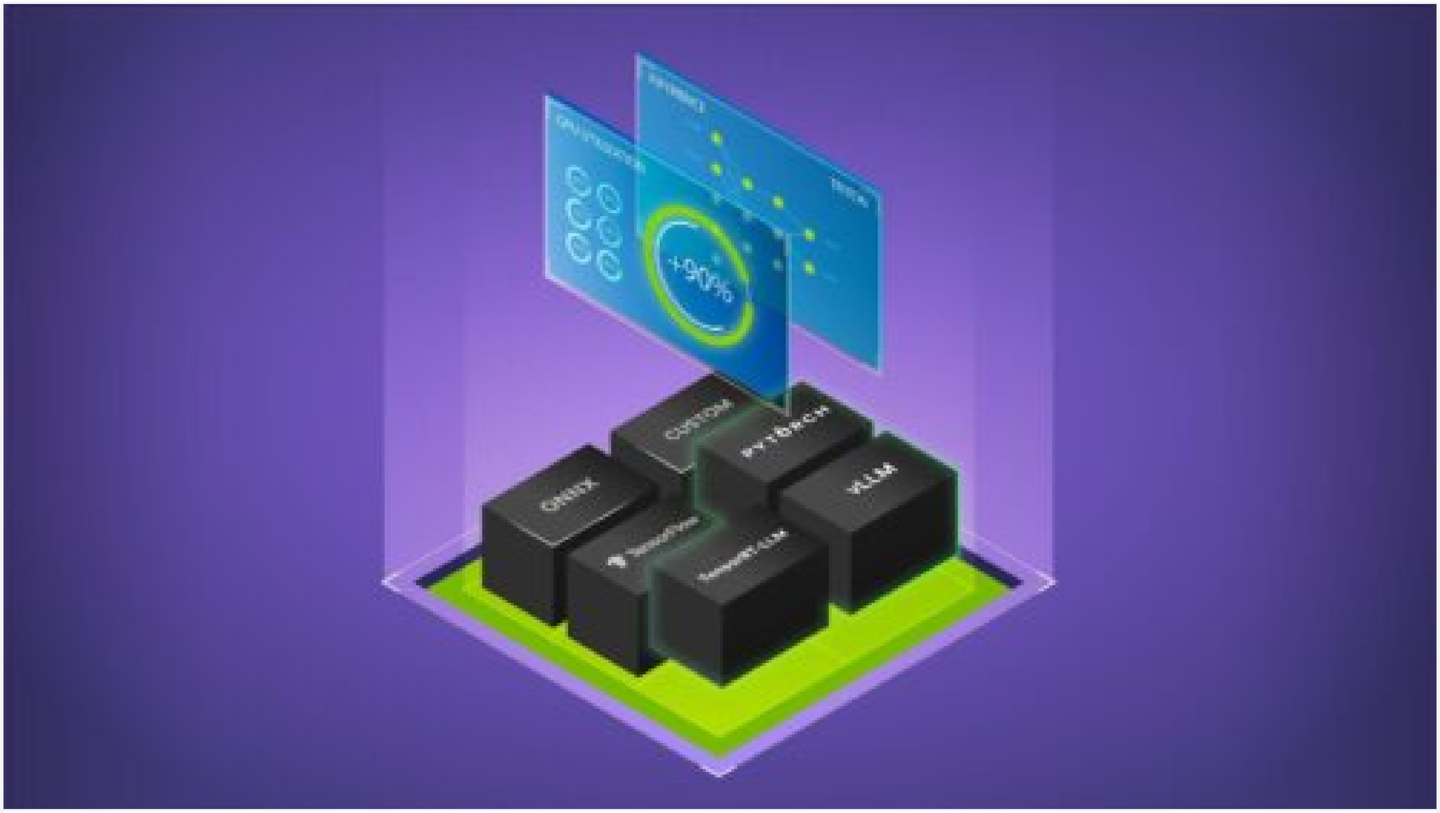


Technical Blog Subscribe



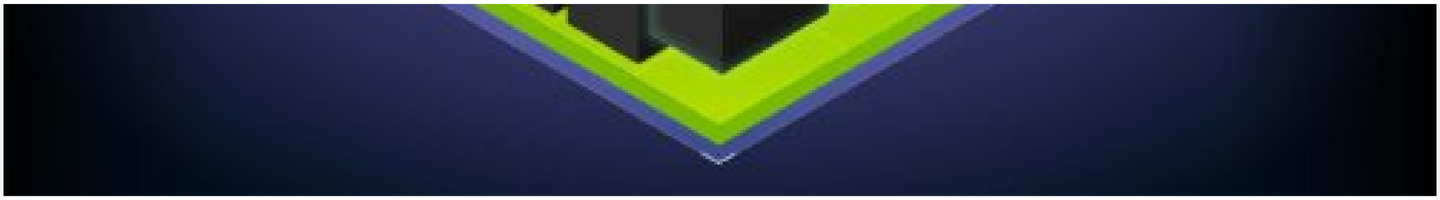
Optimize AI Inference Performance with NVIDIA Full-Stack Solutions

Introduction



NVIDIA Triton Inference Server Achieves Outstanding Performance in MLPerf Inference 4.1 Benchmarks

poor throughput-latency performance and no support for multiple frameworks. NVIDIA Triton



Identifying the Best AI Model Serving Configurations at Scale with NVIDIA Triton Model Analyzer

you're ready to deploy to a Kubernetes cluster. For enterprises looking to trial Triton Inference

Furthering NVIDIA Performance Leadership with MLPerf Inference 1.1 Results

- **SDK:** PyTorch Geometric(PyG) Container
- **Webinar:** Building and Running an End-to-End Machine Learning Workflow, 5x Faster

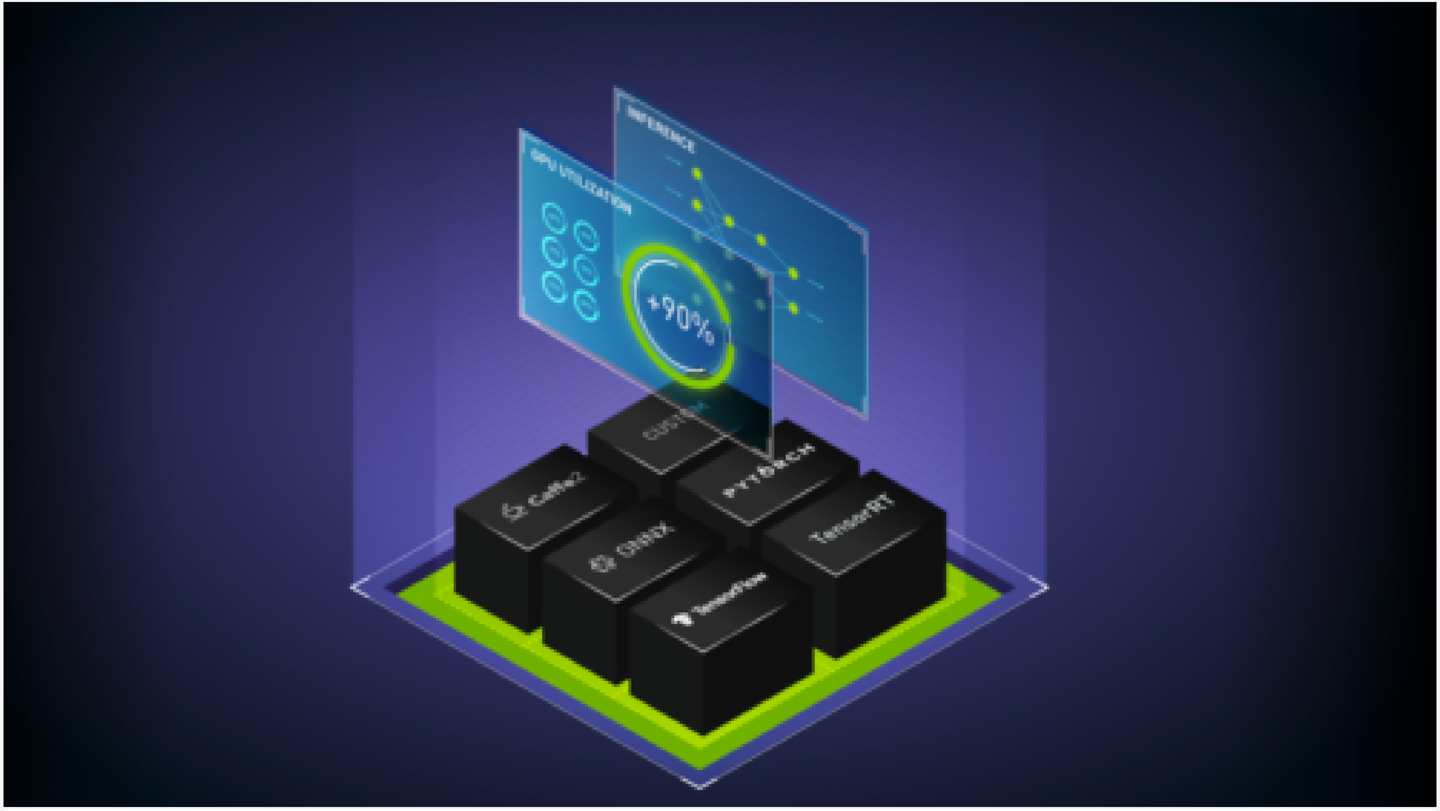
Discuss (0) +4 Like

Tags

About the Authors

About William Hicks

Will Hicks is a Senior Software Engineer on the NVIDIA RAPIDS team. Hicks has an MS in Physics from Brandeis University and an MFA in Literary Arts from Brown University.



Deploying AI Deep Learning Models with NVIDIA Triton Inference Server