Fine-Tuning and Deploying LLMs with NeMo Framework and Triton Inference Server

Emiliano Molinaro, Ph.D.

SDU eScience Center

UCloud Hands-On Workshop, 28-05-2025

SDU eScience

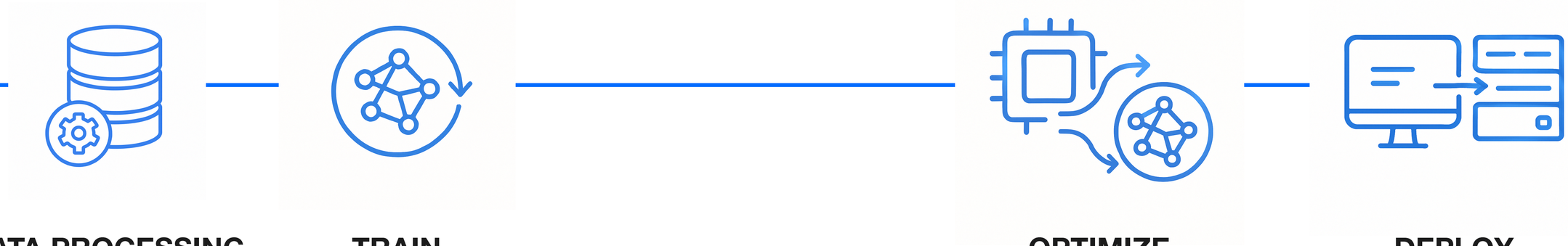
From Training Pipelines to Real-Time Inference

AI Training

AI Inference

NVIDIA NEMO

NVIDIA TRITON INFERENCE SERVER



DATA PROCESSING

collect, cleanse and transform training dataset

TRAIN

select model and fine-tune for desired accuracy

OPTIMIZE

optimize model for target hardware accelerator

DEPLOY

serve model for optimal user inference

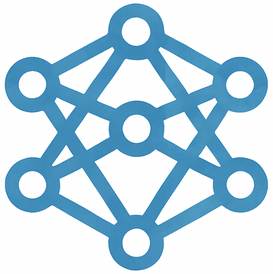
SDU eScience

Overview of NeMo Framework

Performance & Scalability

- Data Parallelism
- Tensor Parallelism
- Pipeline Parallelism
- Context Parallelism
- Expert Parallelism
- GPU Acceleration

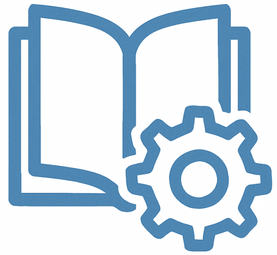
Model Coverage



Supports Several Model Families:

LLMs, SSMs, MoEs, SD, VLMs, …

Training Algorithms



- PEFT: LoRA, P-tuning, IA3, QLoRA, Adapters
- Model Alignment: RLHF PPO, DPO, KTO, IPO, RLAIF, SteerLM, Rejection Sampling

Usability

- Python-based API
- Config-Driven
- Modular
- Production-Ready

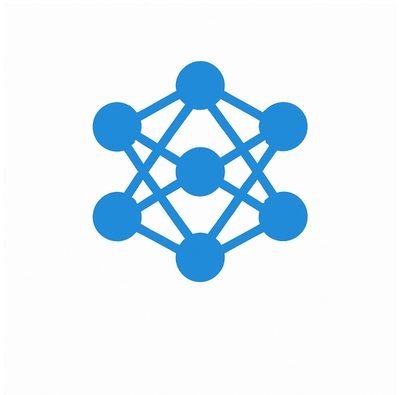
Parameter-Efficient Fine-Tuning

Efficient Fine-Tuning Methods

Technique	How it works	Parameters trained
Prompt Tuning	🔷 Add prompts to input	✅ Prompt embeddings only
Prefix Tuning	🔷 Insert prompts at every layer	✅ Layer-wise prompts only
P-Tuning	🔷 Generate prompts with encoder (LSTM)	✅ Small encoder only
LoRA	🔷 Low-rank adaptation in attention layers	✅ Tiny subset of layer parameters

Overview of Triton Inference Server

Any Framework



Multiple Backends:
TensorFlow, PyTorch,
TensorRT, XGBoost,
ONNX, Python,
TensorRT-LLM, vLLM

Any Query Type

Optimised for Real Time,
Batch, Audio/Video Streaming,
Ensemble Inferencing

High Performance

Model Analyser for
Optimal Configuration

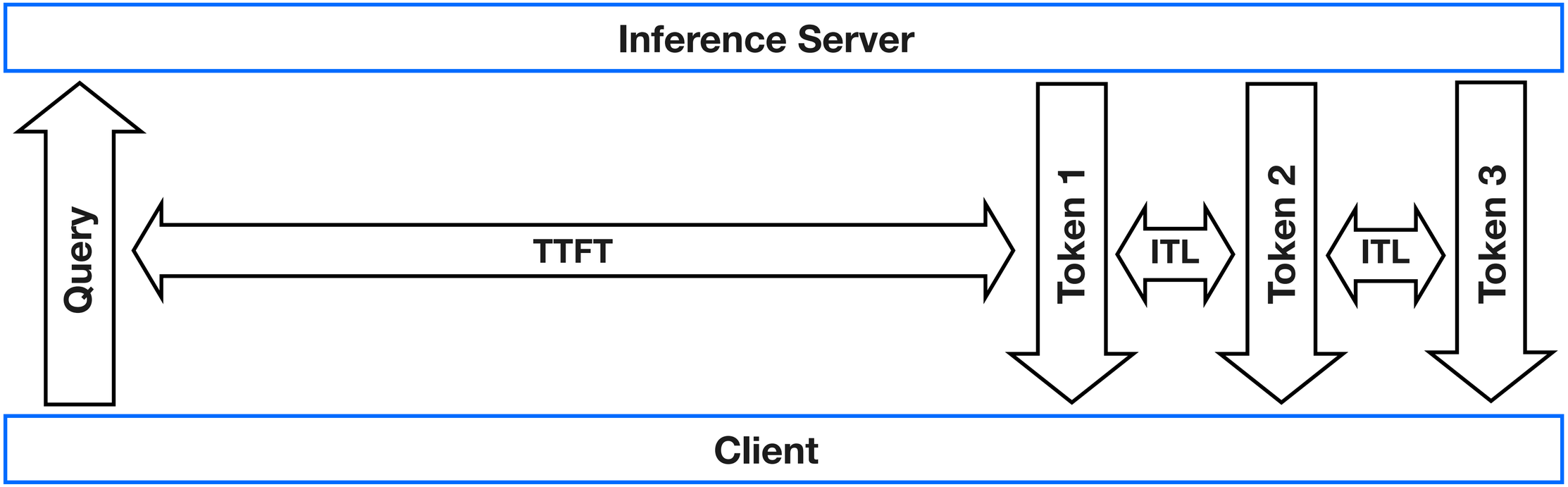
Optimised for High GPU/CPU
Utilisation, High Throughput,
Low Latency

Latency Metrics

- **TTFT (Time To First Token):**
  The duration from the request initiation to the generation of the first token in the response.

- **End-to-End Latency (E2E):**
  The total time taken by the model to generate the entire response from start to finish.

- **Inter-Token Latency (ITL) / Time Per Output Token (TPOT):**
  The average time between the generation of consecutive tokens in streaming scenarios, reflecting client-side token wait times.



SDU eScience

Inflight Batching

Maximize GPU Utilisation during LLM Serving

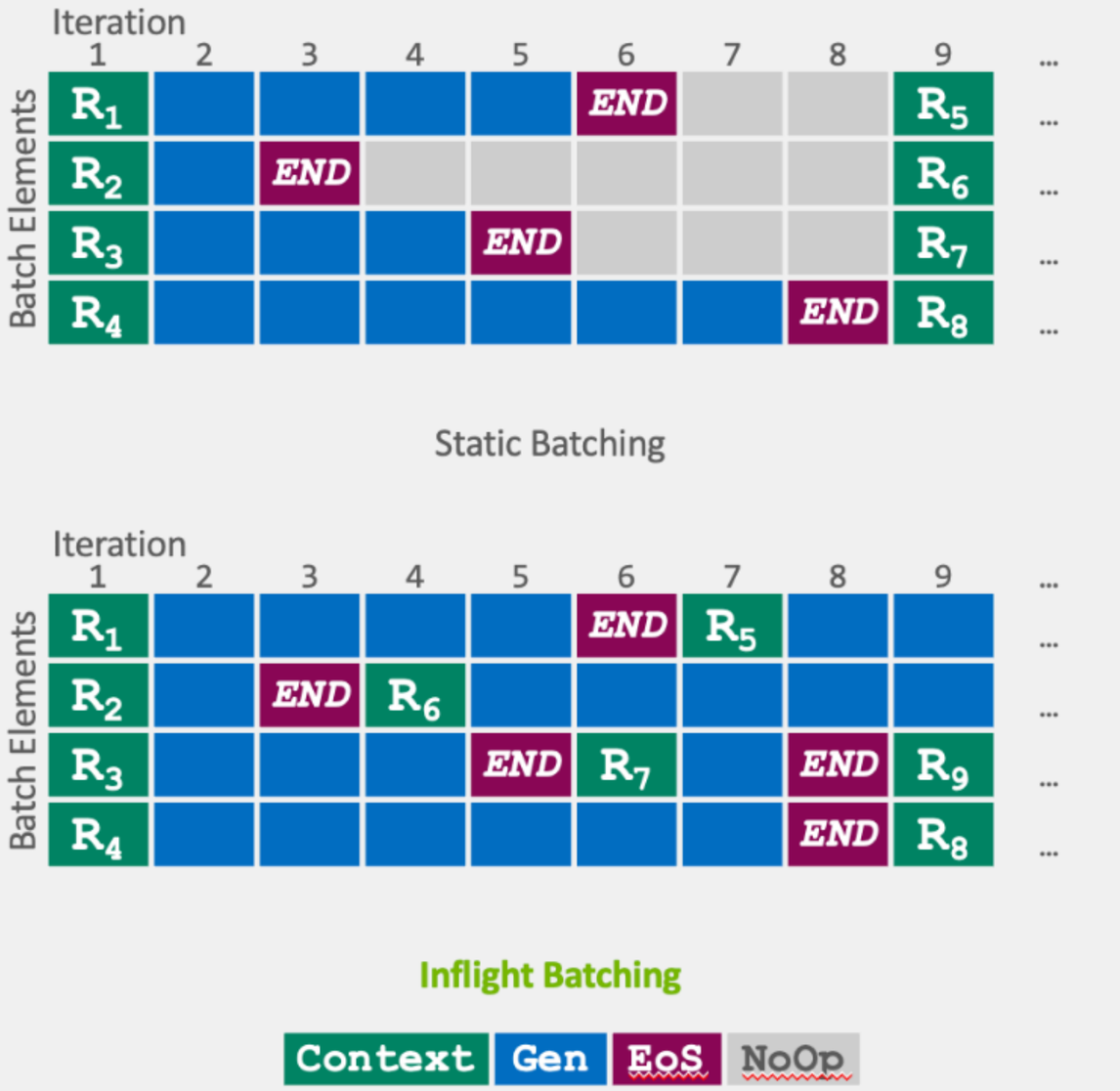
Benefits of Inflight Batching:

- **Enhanced Throughput:**
  - Processes more requests per GPU operation.
  - Reduces per-request overhead, leading to higher overall throughput.

- **Optimized Resource Utilization:**
  - Maximizes GPU usage by maintaining high occupancy rates.
  - Cost-effective scaling by leveraging existing hardware efficiently.

- **Minimal Latency Impact:**
  - Configurable batching parameters ensure that latency remains within acceptable bounds.
  - Balances between batch size and response time to meet application needs.

Figure from NVIDIA



Inflight batching and the additional kernel-level optimizations improve GPU usage and doubles throughput which helps reduce energy costs and minimize TCO.

Hands-On Session

Implement end-to-end workflow on **UCloud - Interactive HPC**:

- Deploying an LLM in production for optimized inference
- Creating a synthetic medical Q&A dataset
- Setting up and fine-tuning a LLM with PEFT (LoRA)

Notebooks available in this GitHub repository:

https://github.com/emolinaro/ucloud-workshop-28-05-2025