SNIA | Data, Storage & Networking

AI Model Inferencing and Deployment Options

Live Webinar
November 13, 2025
10:00 am PT / 1:00 pm ET

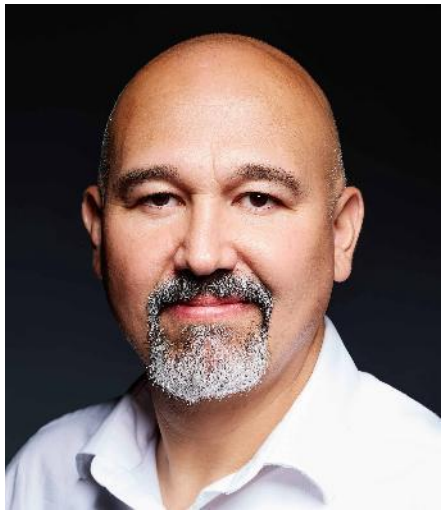
Kevin Marks
Dell Technologies

Luis Freeman
IBM

Tim Lustig
NVIDIA

Today's Presenters

Tim Lustig
Director, Corporate Ethernet Marketing
NVIDIA
Moderator



Luis Freeman
Technical Project Manager - Technology and Strategy for Storage Devices
IBM

Kevin Marks
AI Technologist / Distinguished Engineer
Dell Technologies

The SNIA Community

200
industry leading organizations

2,000
active contributing members

50,000
IT end users & storage pros worldwide

SNIA | DSN A SNIA Community

SNIA | Data, Storage & Networking

What We Do

Drive the awareness and adoption of a broad set of technologies, including:

- Storage Protocols (Block, File, Object)
- Traditional and software-defined storage
- Disaggregated, virtualized and hyperconverged
- AI, including storage and networking considerations
- Edge implementation opportunities and factors
- Storage and networking security
- Acceleration and offloads
- Programming frameworks
- Sustainability

How We Do It

By delivering:

- Expert webinars and podcasts
- White papers
- Articles in trade journals
- Blogs
- Social Media
- Presentations at industry events

Logistics

- The slides are available under the attachments tab at the bottom of your console.
- Questions are welcome!
- Please rate the session and provide feedback!
- Want more sessions like this or other topics, let us know!
  - JOIN US! We meet on Thursday mornings at 11:00 AM eastern.
  - Email dsn-chair@snia.com if you have questions.

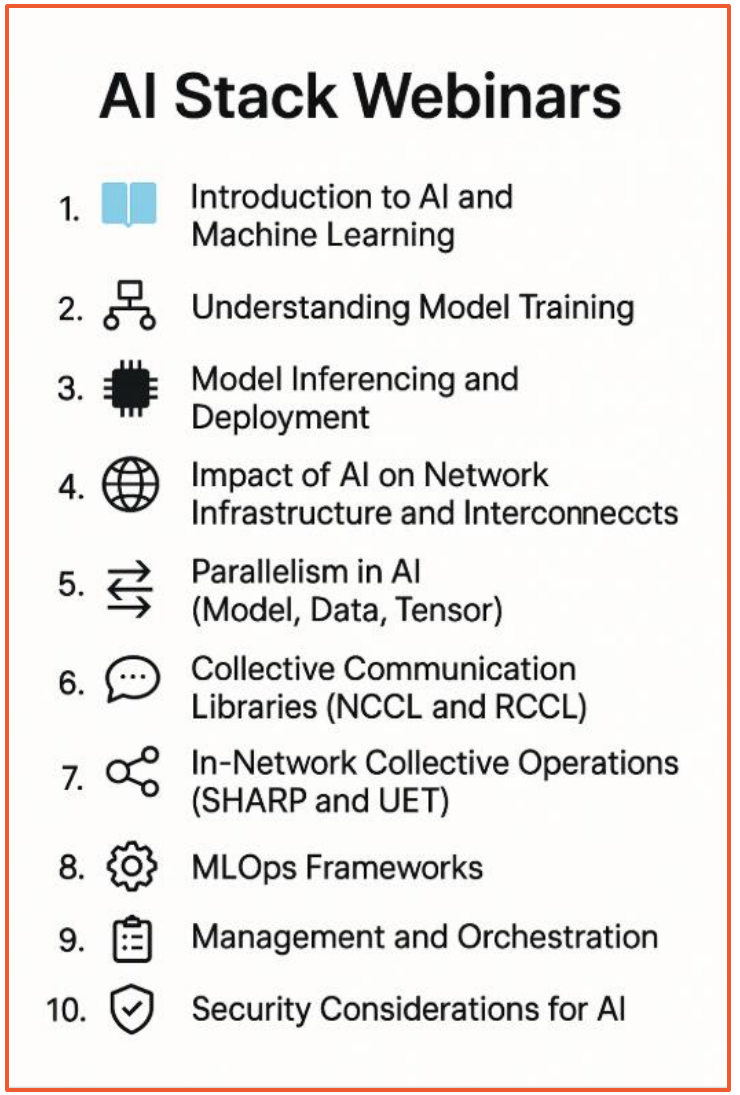
SNIA Legal Notice

- The material contained in this presentation is copyrighted by SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of SNIA.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

  NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

The “AI Stack” Webinar Series

- Building a Strong Foundation for All Experience Levels:
  - Starting from the basics
  - Building steps-by-step
  - Connecting theory to practice
  - Demonstrations
  - Preparing for real-world challenges



Agenda

- Deep Learning Models
- AI Model Inference
- AI Model Serving (Deployment)
- LLM Model Inferencing Basics (with Demos)
- Q&A

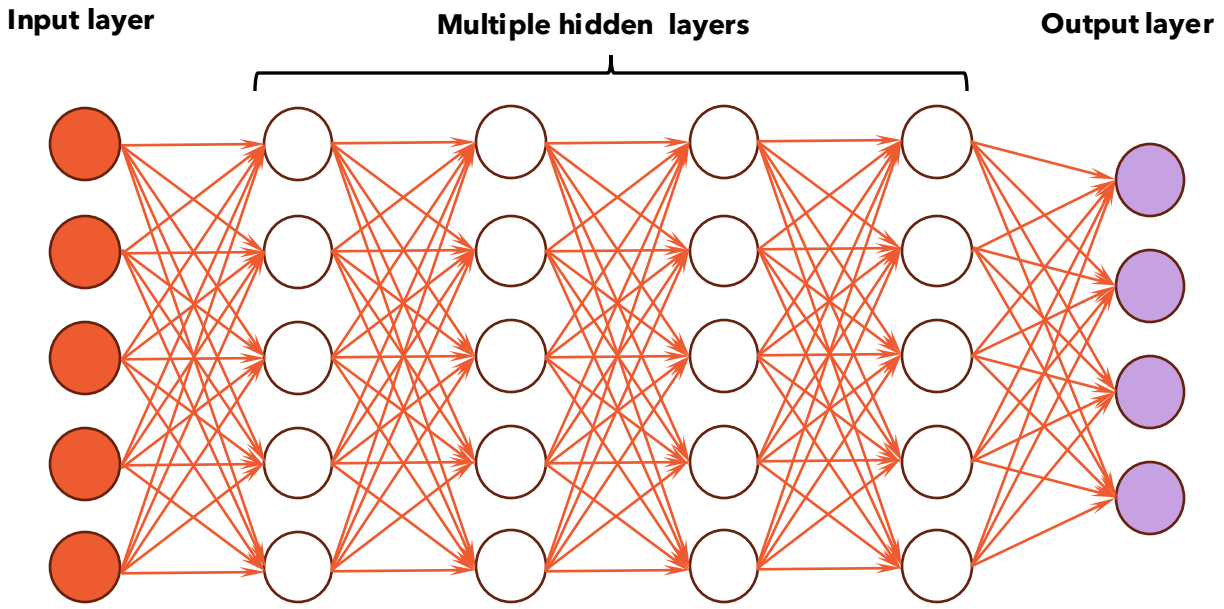
SNIA | DSN
A SNIA Community

Deep Learning Models

What are Deep Learning (DL) Models

Deep Learning Models are multi-layered neural networks consisting of an input layer, multiple hidden layers and an output layer.

Architectural design changes to the hidden layers define the different types of Deep Learning Models. Convolutional filters, pooling layers and feedback loops are a few examples of these architectural design changes.



Deep Learning Model Types

Generative Models (Gen AI)

- Can create original content (text, images, video, audio, code) in response to a user's request.
- Unsupervised Learning:
  Uses 2 neural nets working together or in opposition to learn.
- Examples of Gen-Ai models
  - Variational autoencoders (VAEs)
  - Generative Adversarial Networks (GANs)
  - Diffusion models
  - Transformers

Transformers are the basis for modern AI models like ChatGPT, Claude, Llama, Gemini, Granite.

Non-Generative Models

- Earlier models, focus is on analyzing and understanding existing data
- Supervised Learning:
  Require labeled data to learn.
- Examples of Non-Gen Models
  - Convolutional Neural Networks (CNNs)
  - Recurrent Neural Networks (RNNs)
  - Long Short-Term Memory (LSTMs)
  - Autoencoders

The Importance of the Transformer Architecture

The transformer overcame the limitations of previous NLP models to process sequential data by introducing the following features:

- Self-attention mechanisms

  Self-attention allows the AI model to calculate the relationships and dependencies between tokens, especially ones that are distant from one another in the text by examining an entire sequence simultaneously and making decisions about what tokens to focus on.

- Parallelism:

  Enables transformers to scale massively and handle complex NLP tasks by building larger models.

These qualities allowed transformers to handle unprecedently large datasets, allowing the recent advances in AI.

Modern models using Transformers:

- Large Language Models (LLMs):

  Can understand and generate human-like text by learning from massive amounts of data.

  Applications: translation, summarization, question answering and content creation.

- Vision Language Models (VLMs):

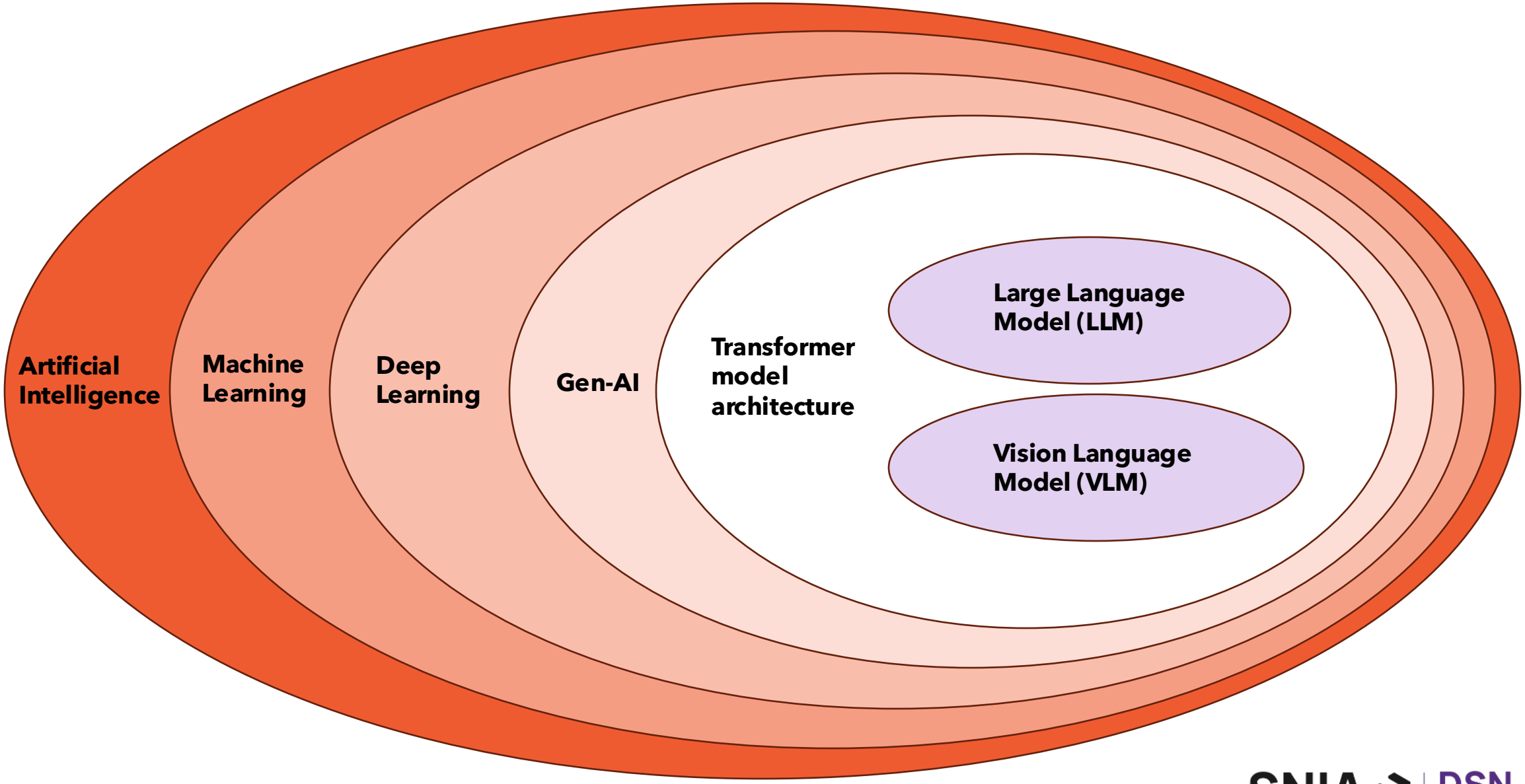
  Are multimodal models capable of understanding and processing both visual and text data.

  VLMs have 2 components:

  - Language Encoder → Creates Word Embeddings.
  - Vision Encoder → Uses a Vision Transformer (ViT) to create Feature Vectors that in turn are converted into Word Embeddings.

  Applications: Captioning, Image Search and Retrieval, Image Generation, Image Segmentation, Visual Question answering.

Relationship AI ←→ LLMs/VLMs



SNIA | DSN A SNIA Community

SNIA | DSN
A SNIA Community

AI Model inference

The Lifecycle of ML

This webinar's Demo will focus on LLM inference

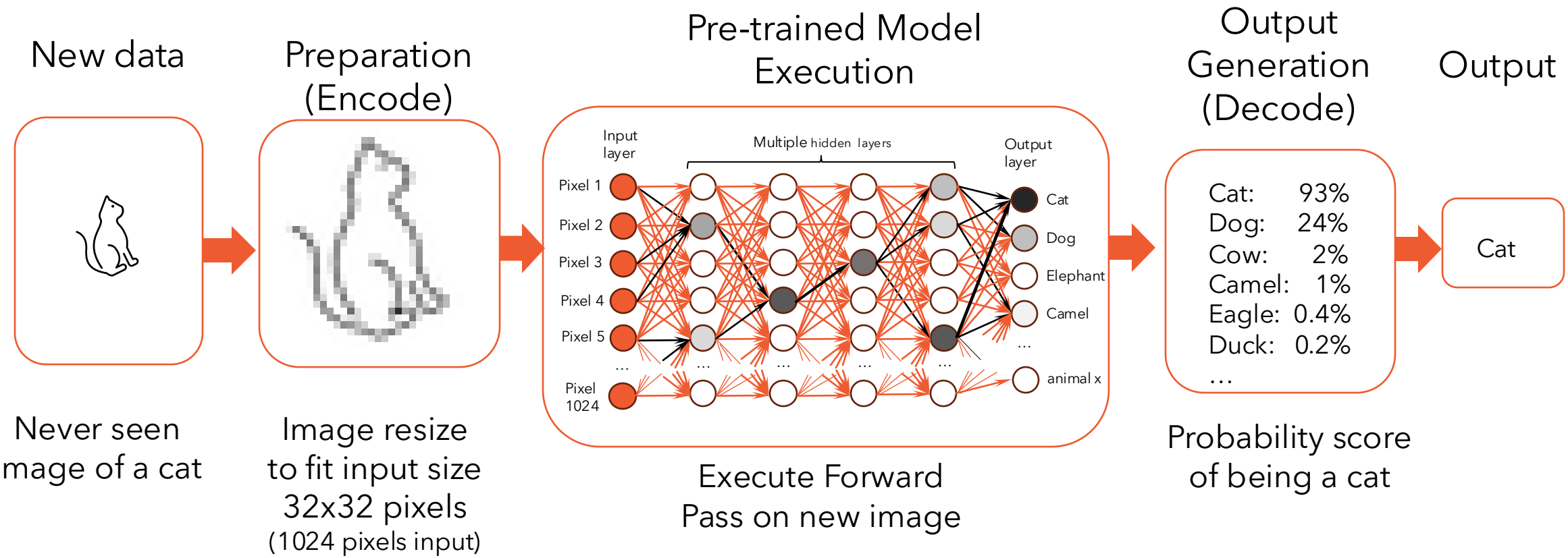
Training	Fine-tuning	Inference	Serving
Builds a new model from scratch.	Adapts a pre-trained model to specific task.	Uses trained model to make predictions.	Deploys and manages model to handle inference requests.
Learns from a large dataset. LLMs ➔ ~10TB text. CNNs ➔ ~50GB images	Refines existing model. LLMs ➔ Q&A dataset. CNNs ➔ selected images	Uses new (unseen) data.	Packages and exposes the model as an API.
Requires a cluster of GPUs LLMs ➔ ~6,000 x 12 days CNNs ➔ 1 GPU x few hrs.	Runs for few hours on small GPU cluster. LLMs ➔ Few GPUs x days CNNs ➔ 1 GPU x few hrs	Response is instantaneous.	Response time depends on use case

Machine Learning Operations (MLOps)

What is Inference – Computer Vision

Inference in Artificial Intelligence is the process where a pre-trained AI model uses its learned knowledge to analyze new, unseen data to generate predictions or conclusions.

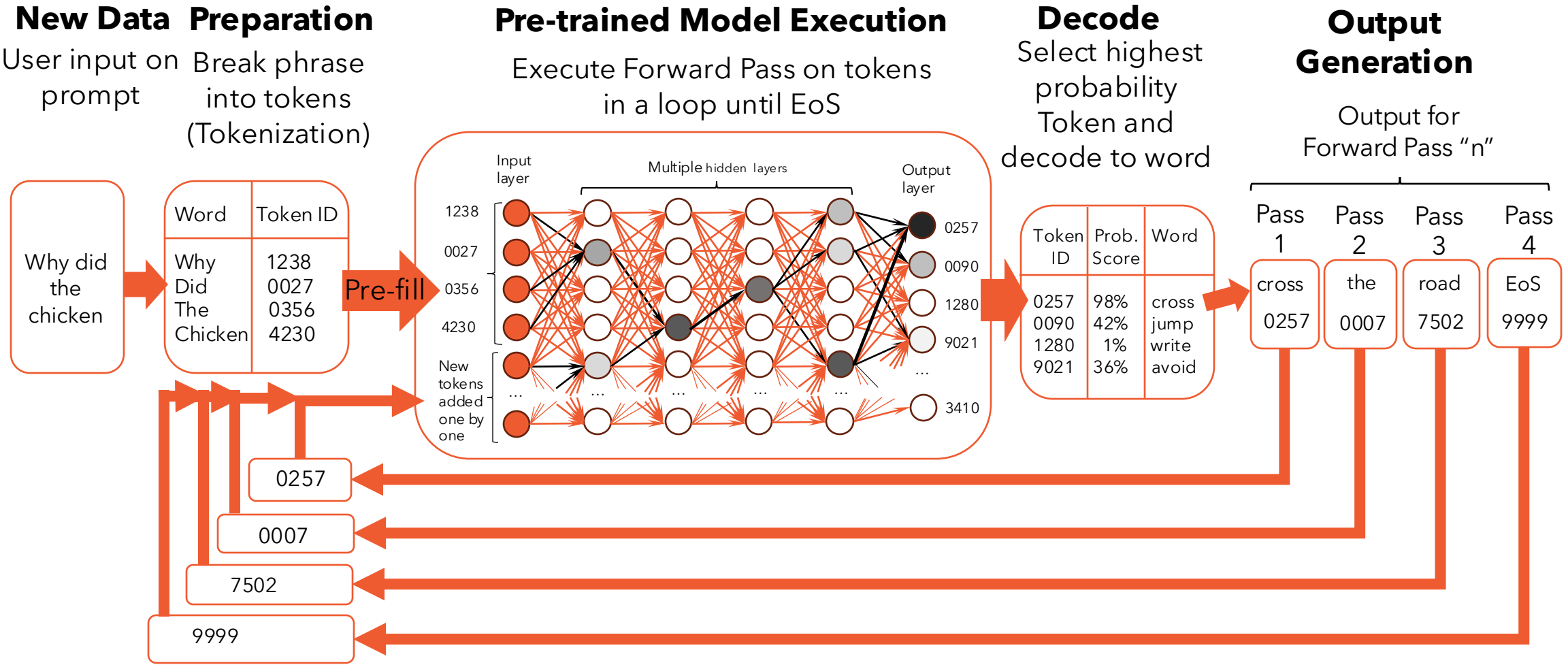
Let's use image classification as an example.



SNIA DSN A SNIA Community

What is Inference – Natural Language (LLMs)

Inference for Natural Language Processing is autoregressive, meaning it runs on a loop predicting the next word and adding it to the previous input until it reaches an End of Sequence (EoS). LLM inference is computationally costly.



SNIA DSN A SNIA Community

Challenges and Optimization of LLM Inference

Challenges

- **Slow:** LLM Inference generates 1 output token at a time.
- **Costly:** With every new inference a full Forward Pass is needed. That's billions of parameters loaded to GPU and computed at each inference pass.
- The bottleneck is memory bandwidth, not GPU compute power.

Solutions for optimization

- **Model Size Optimization:** Quantization of model by reducing data type of model weights. Reduces memory requirements and speeds up computation.
- **Batching and Caching:** like Key Value (KV) cache, which allows to compute attention on the latest token without having to re-load and compute all model weights for previous tokens.
- **Accelerator parallelism:** Data or model is split across several GPUs
- **Architectural Optimizations:** like speculative decoding uses small draft model to guess (speculate) the next "n" tokens and then validated all "n" tokens in one pass using large target model.
- **Numeric Optimizations:** Custom Kernels
- **Inference frameworks** like vLLM, TensorRT-LLM, Ollama, SGLang, etc

SNIA | DSN
A SNIA Community

AI Model Serving (Deployment)

What is Model Deployment

Model deployment involves placing a Machine Learning (ML) model into a production environment. Moving a model from development into production makes it available to end users.

Depending on the application and use case for the model, one of the following deployment methods can be used.

On the Cloud / On-Prem

Most common approach where inference runs on powerful remote servers in a datacenter.

Benefits:

- **Scalability:** Unlimited resources makes it ideal for scalability
- **Powerful:** Can handle massive datasets and complex models.
- 2 modes of inference:
  - Real-Time (online) inference
  - Batch (offline) inference.

On the Edge

Performs inference directly on the device where the data is generated.

Benefits:

- **Speed:** Inference is nearly instantaneous, which is critical for autonomous applications like vehicles, robots, drones.
- **Privacy:** Sensitive data is processed on-device.
- **Offline:** does not require internet connectivity

Challenges for LLM Deployment

- **Task Composability challenges:**
  Applications with multiple tasks and control flows can fail either with one or more tasks failing or the overall solution succeeding but outcome being incorrect (composability gap).

- **Privacy concerns:**
  How to ensure the model will not leak personal or confidential information.

- **Security challenges:**
  How to ensure model does not leak passwords or give access to system.

- **Hallucination:**
  Poor performance on tasks that require factuality.

- **Data drift:**
  Models trained on data collected in the past fail to generalize to answering questions asked in the present, even when provided with an updated evidence corpus.

Prompt Engineering Challenges

- **Consistency:**
  Stochastic output of LLMs causes different output to the same prompt input. No output schema is guaranteed.

- **Prompt Versioning:**
  Small changes in a prompt can lead to different results, making it essential to track and version each prompt and its performance.

- **Backward and Forwards compatibility:**
  How to make sure your prompts work with new data added to current model or new model.

- **Context Length:**
  Information seeking questions have context-dependent answers. Examples: document processing, Summarization, Narrative, etc.

Large Language Model Operations (LLMOps)

Specialized set of practices and tools for managing the entire lifecycle of Large Language Models.

- **Version control:**
  Track model versions and their related data source and metadata.
  Tools: DVC, Git, Gitlab, Weights&Biases.

- **Data and workflow orchestration:**
  Manages the data pipelines needed for LLMs and creates scalable workflows to ensure efficient performance.
  Tools: Kubernetes, OpenShift, ECS.

- **Model training and fine-tuning:**
  Addresses the complex process of training and adapting pre-trained models for specific tasks.
  Tools: TensorFlow, PyTorch.

- **Prompt engineering:**
  Focuses on refining and managing the prompts that are sent to the model, as well as the outputs they generate, to improve quality and control.
  Tools: LangChain, Agenta, PromptFlow, Promptmetheus.

- **Deployment and infrastructure:**
  Handles the technical backbone for deploying LLMs, ensuring they are robust and efficient, and often involves strategies for cost and latency management.
  Tools: Kuberflow, Mlflow, Docker.

- **Monitoring and reliability:**
  Continuously monitors model performance in production to detect issues like slow responses or hallucinations, while also ensuring ethical standards and addressing biases.
  Tools: Prometheus, Grafana.

- **Continuous Integration / Delivery (CI/CD)**
  Applies DevOps principles like automated testing, building and deployment of models.
  Tools: Jenkins, CML

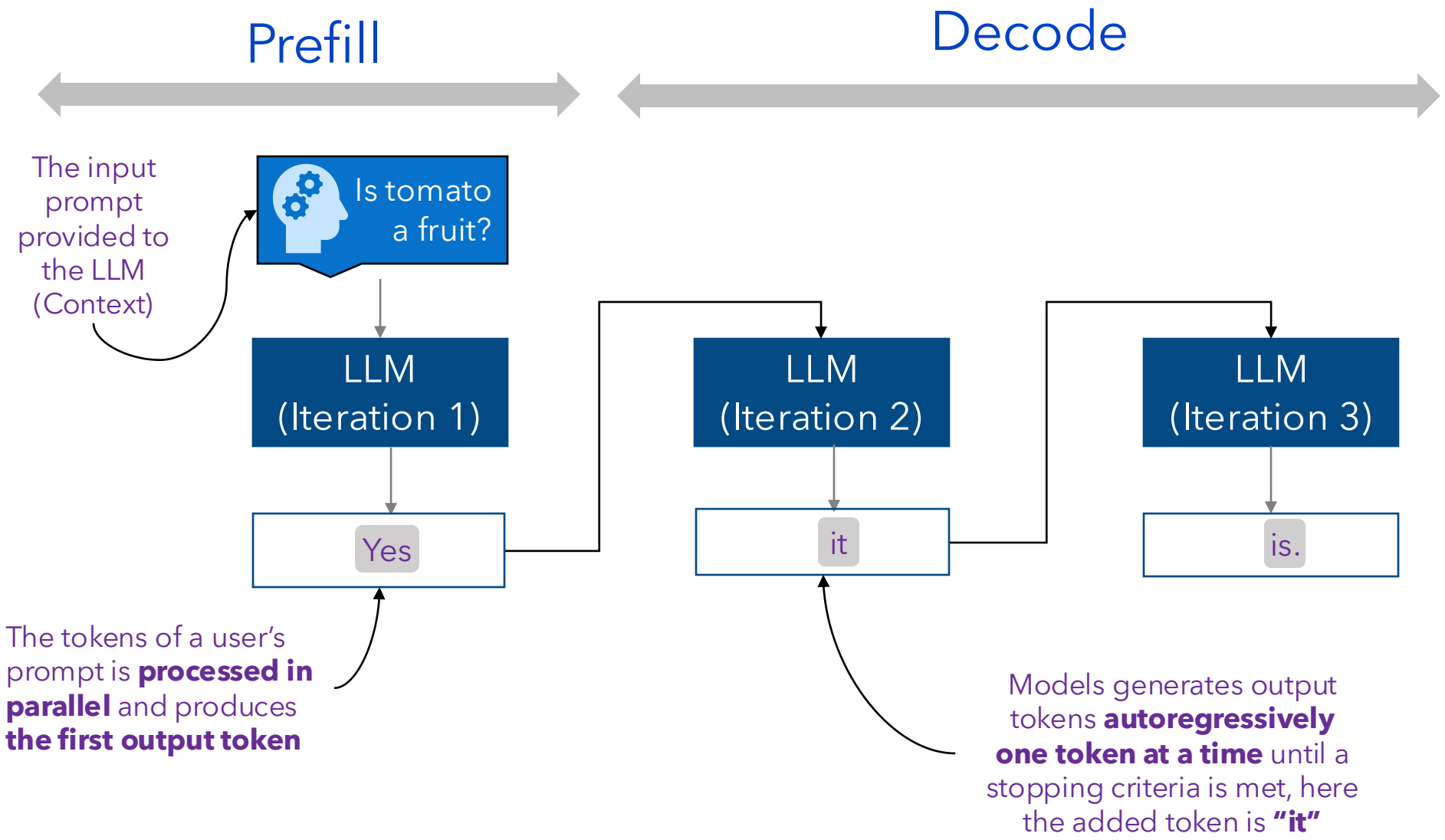
- **Security and compliance:**
  Protects against adversarial attacks and ensures the application adheres to necessary regulations and data security policies.

End-to-End Platforms from Cloud Providers: Amazon SageMaker, Azure ML, Google Vertex AI, IBM Watson Studio

SNIA | DSN
A SNIA Community

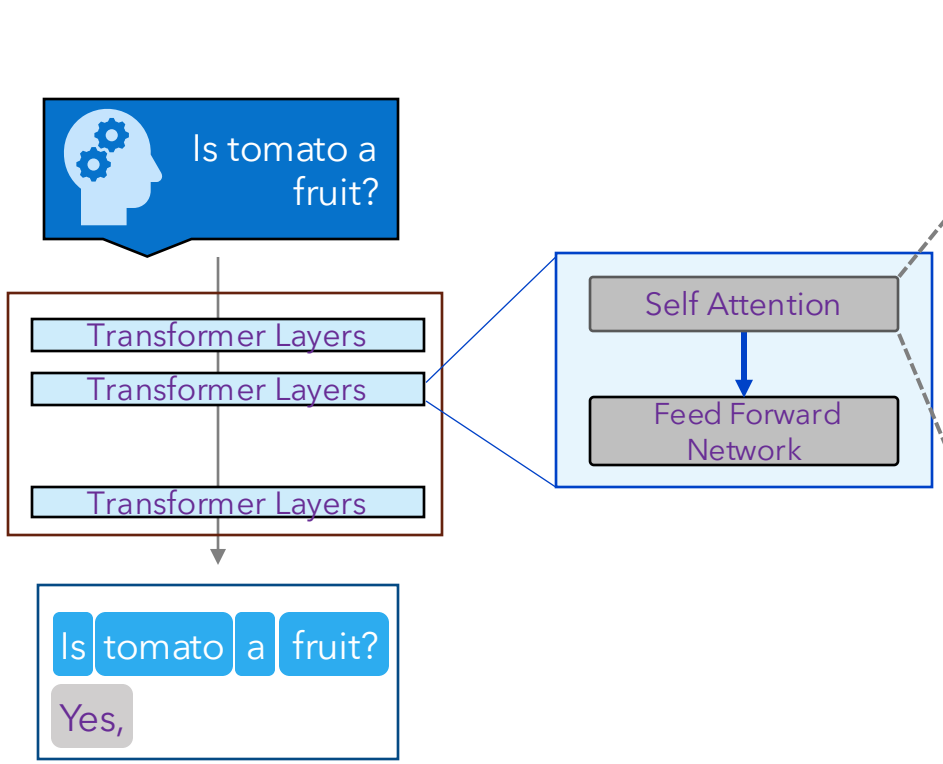
LLM Model Inferencing Basics

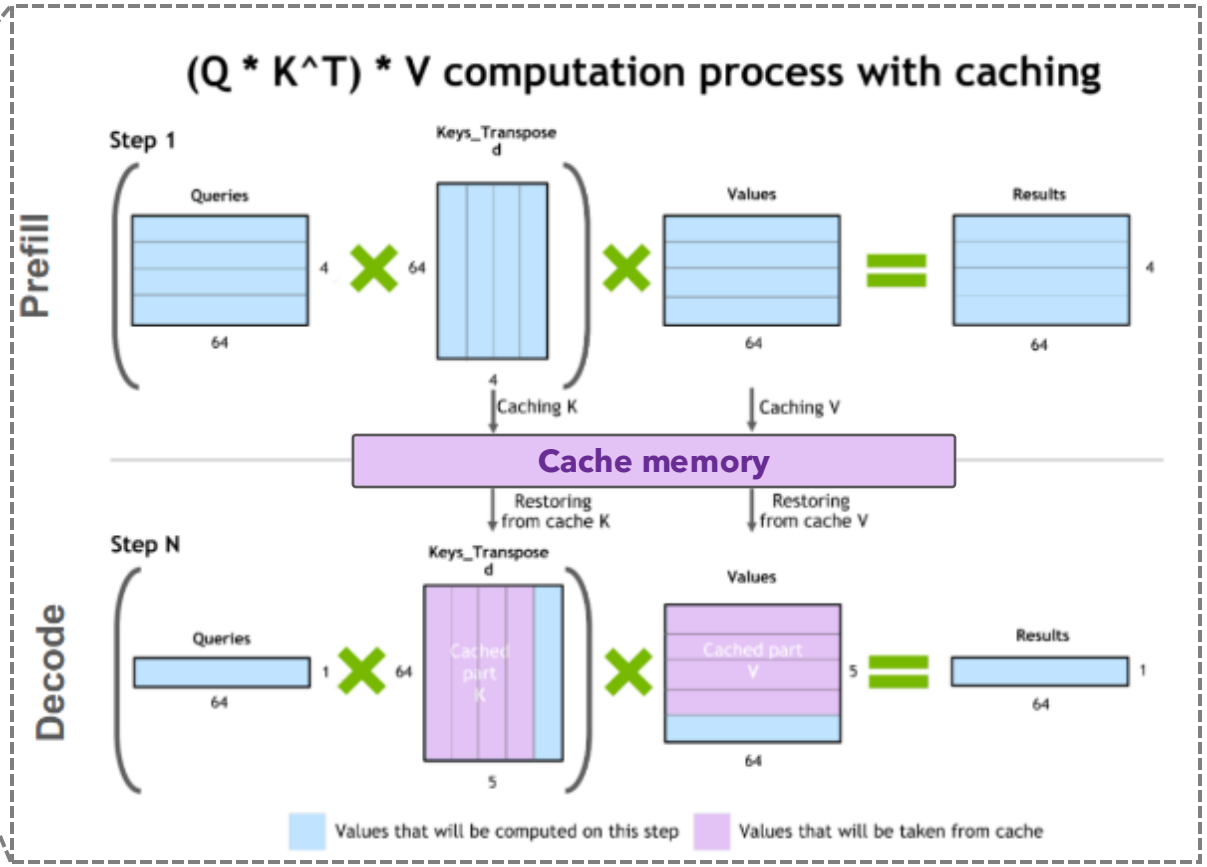
LLM Inference Workflow



Transformer and KV Cache

Instead of recomputing attention for all previous tokens at every step, **KV-Cache stores the key and value vectors** from earlier tokens. It only computes attention for the **new token**.

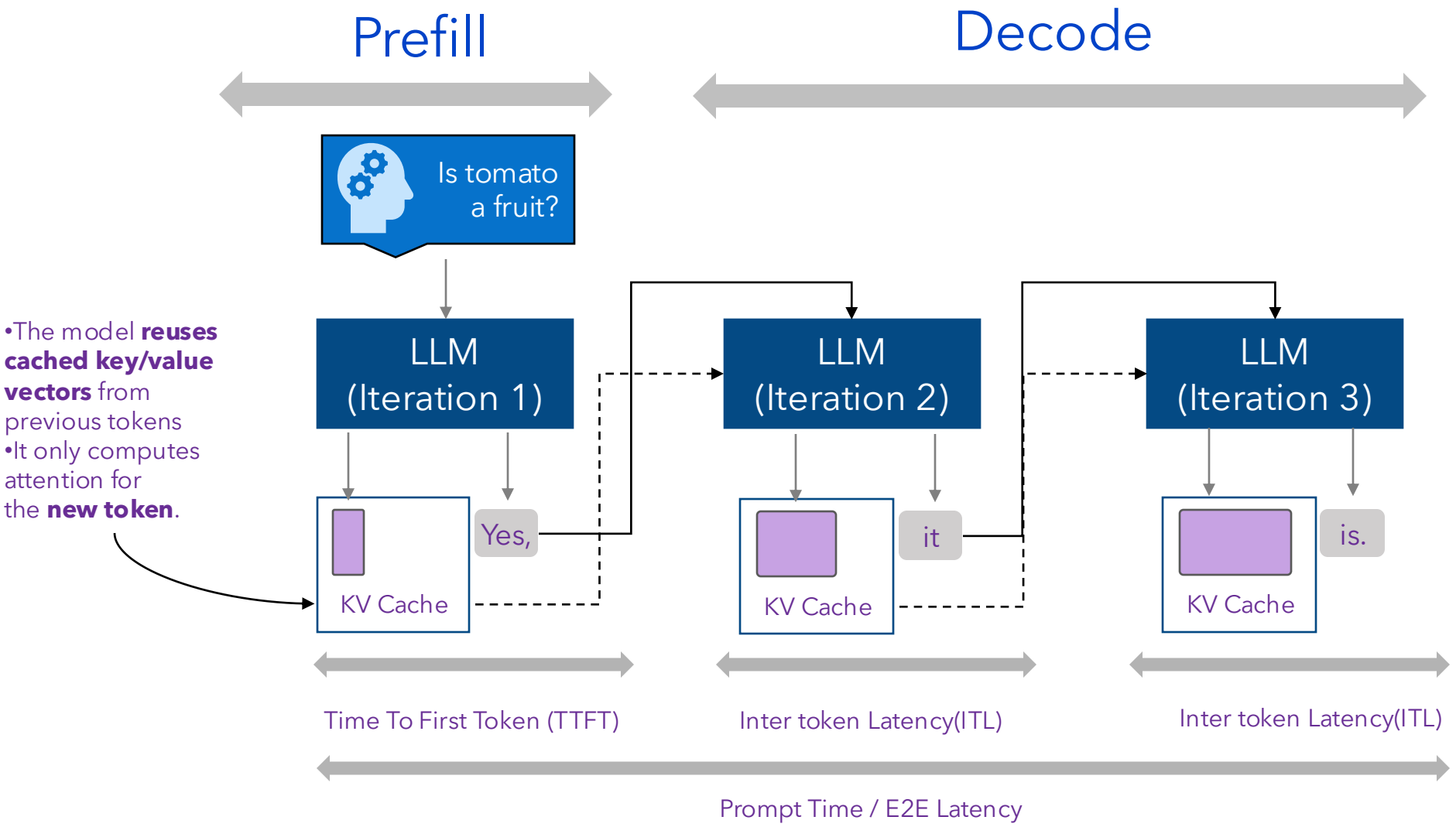




Reference
[1] LayerKV: Optimizing Large Language Model Serving

SNIA DSN A SNIA Community

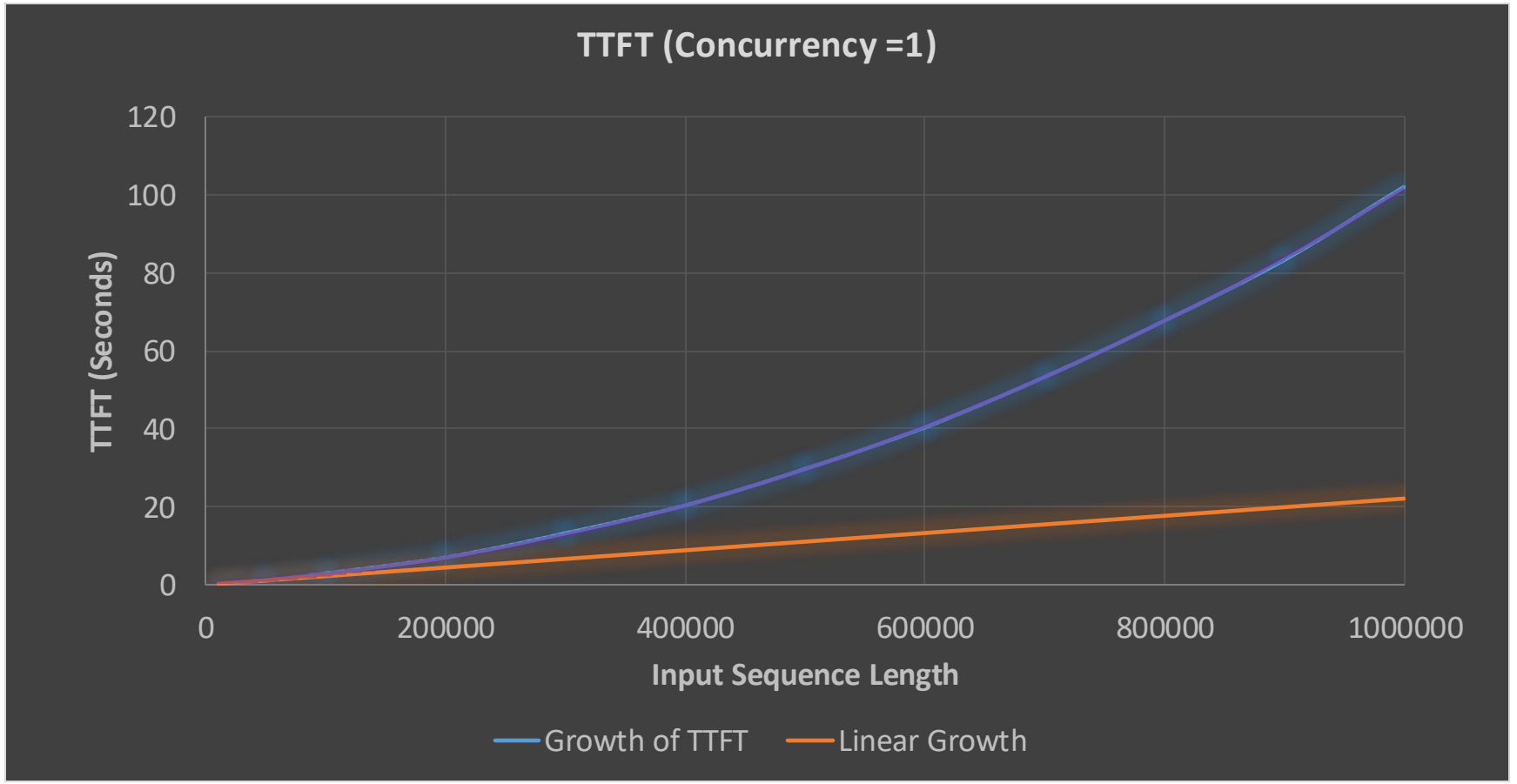
KV Cache is an optimization for transformer based LLM inference



- ***Prefill Phase***: A **compute-bound** task where input tokens (prompt and context) are processed to build KV cache.
- ***Decode Phase:*** A **memory-bound** task where tokens are generated one at a time using KV cache.

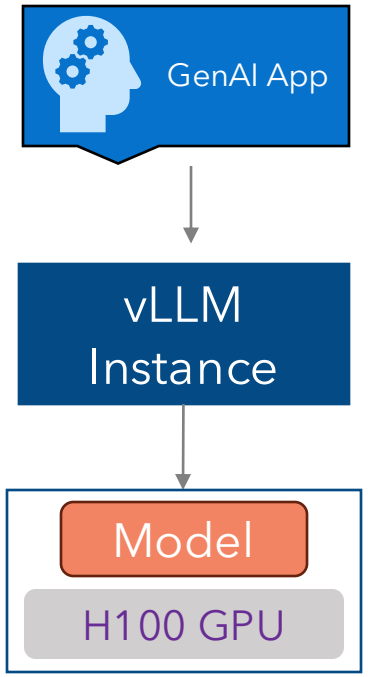
KV-Cache grows and is persisted to the end of the sequence

Growth of TTFT as Input Context Grows



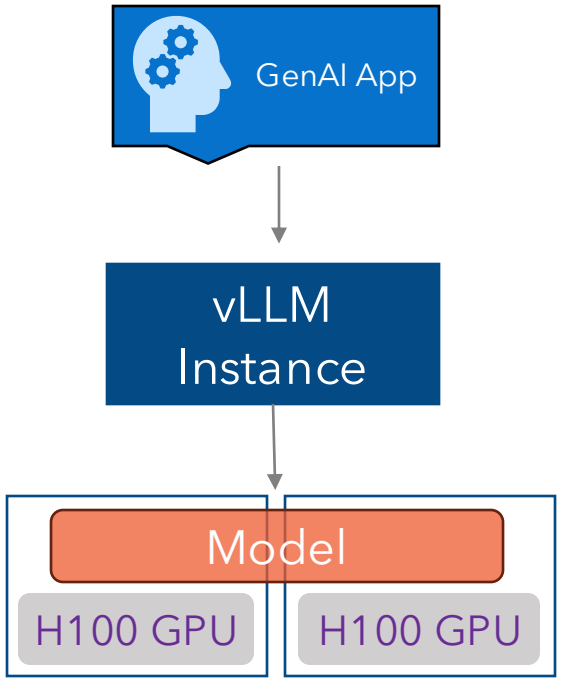
Demo Environment

Port 8001



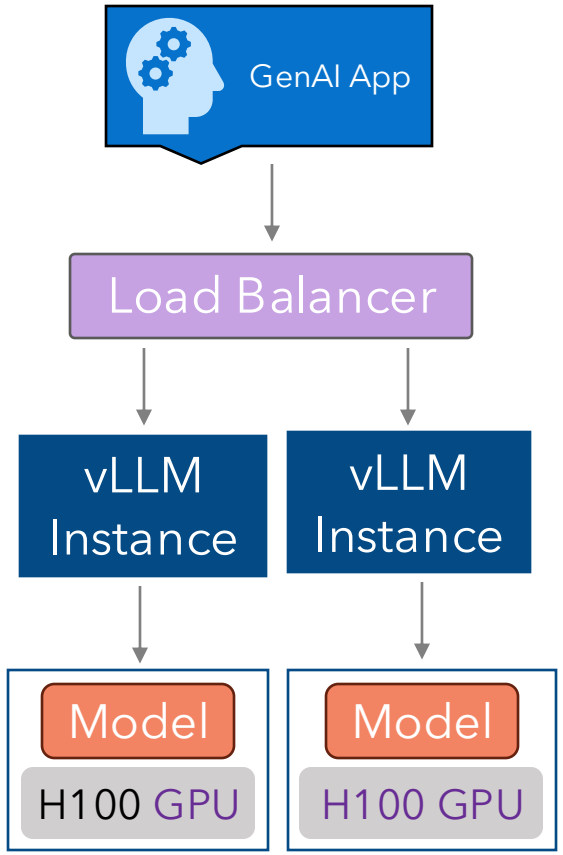
Tensor Parallelism = 1
Pipeline Parallelism=1
Data Parallelism =1

Port 8002



Tensor Parallelism = 2
Pipeline Parallelism=1
Data Parallelism =1

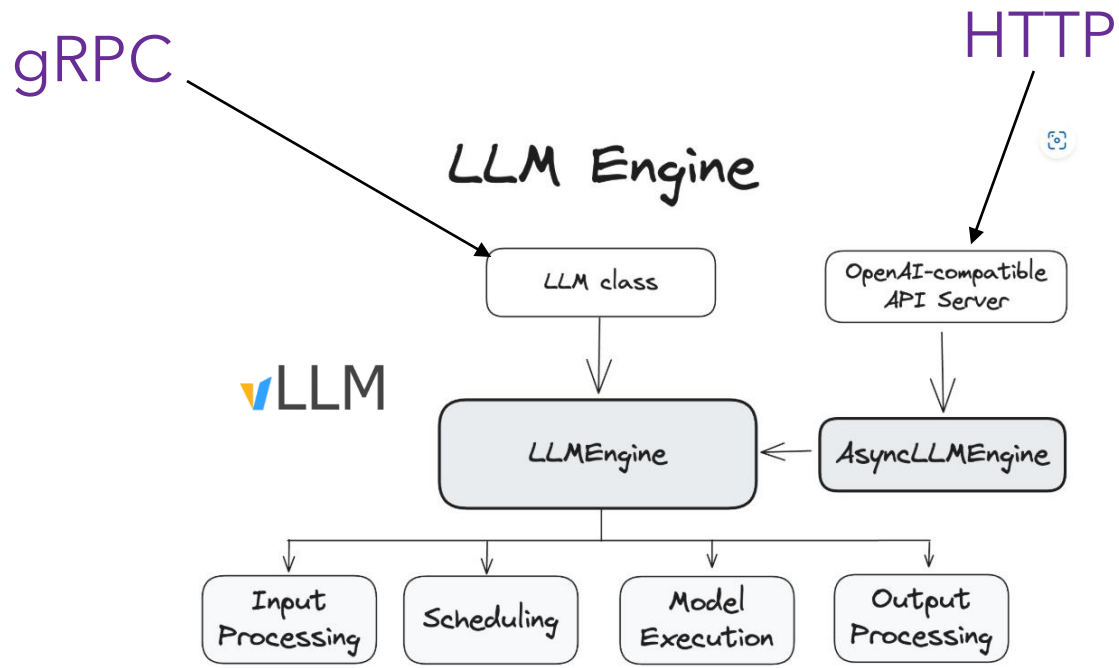
Port 8003



Tensor Parallelism = 1
Pipeline Parallelism=1
Data Parallelism =2

isl – Input Sequence Length | osl – Output Sequence Length | np - Concurrency

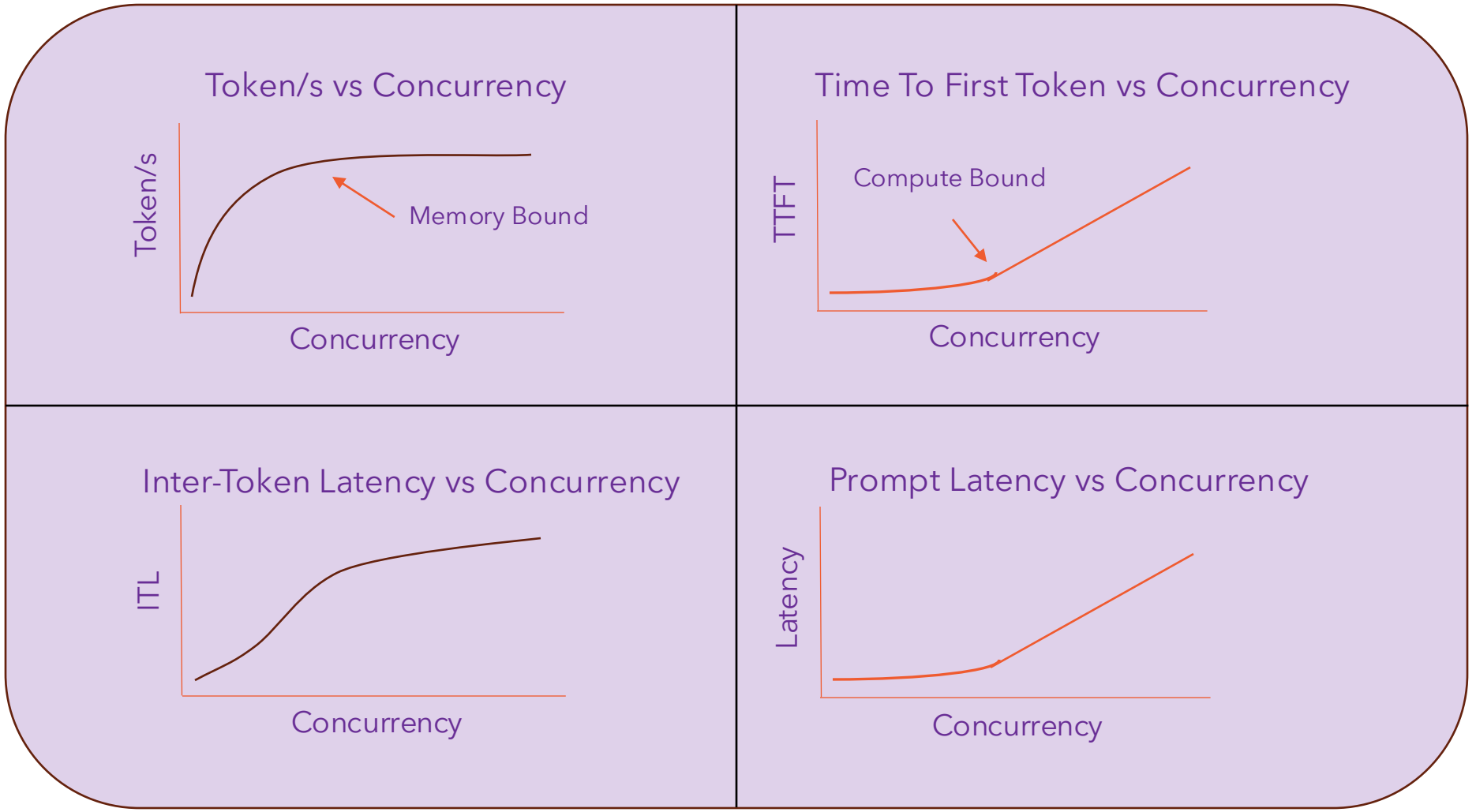
Anatomy of the Inferencing Server



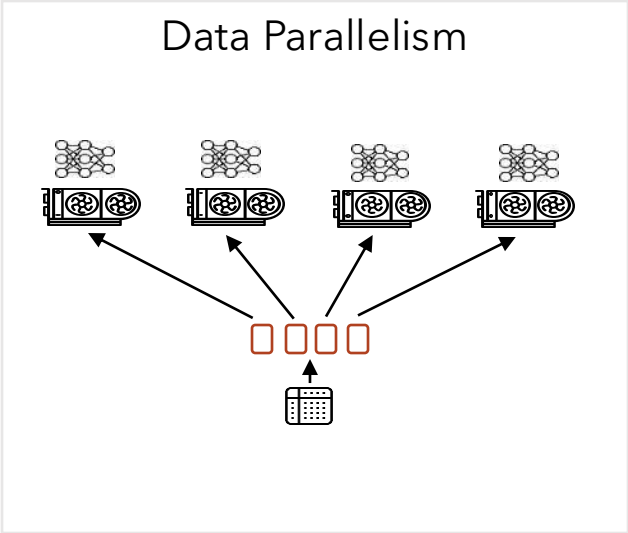
- **Input Processing**: Handles tokenization of input text using the specified tokenizer.
- **Scheduling**: Chooses which requests are processed in each step.
- **Model Execution**: Manages the execution of the language model, including distributed execution across multiple GPUs.
- **Output Processing**: Processes the outputs generated by the model, de-tokenizing back into human-readable text.

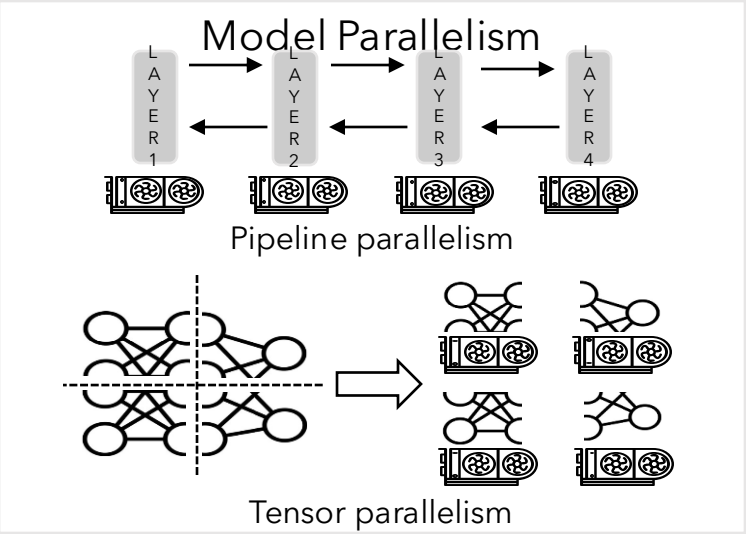
Reference
https://gist.github.com/rbiswasfc/678e4c78258480dcb6214efeedbe5af8

Typical Graph Shapes



Parallelism Techniques

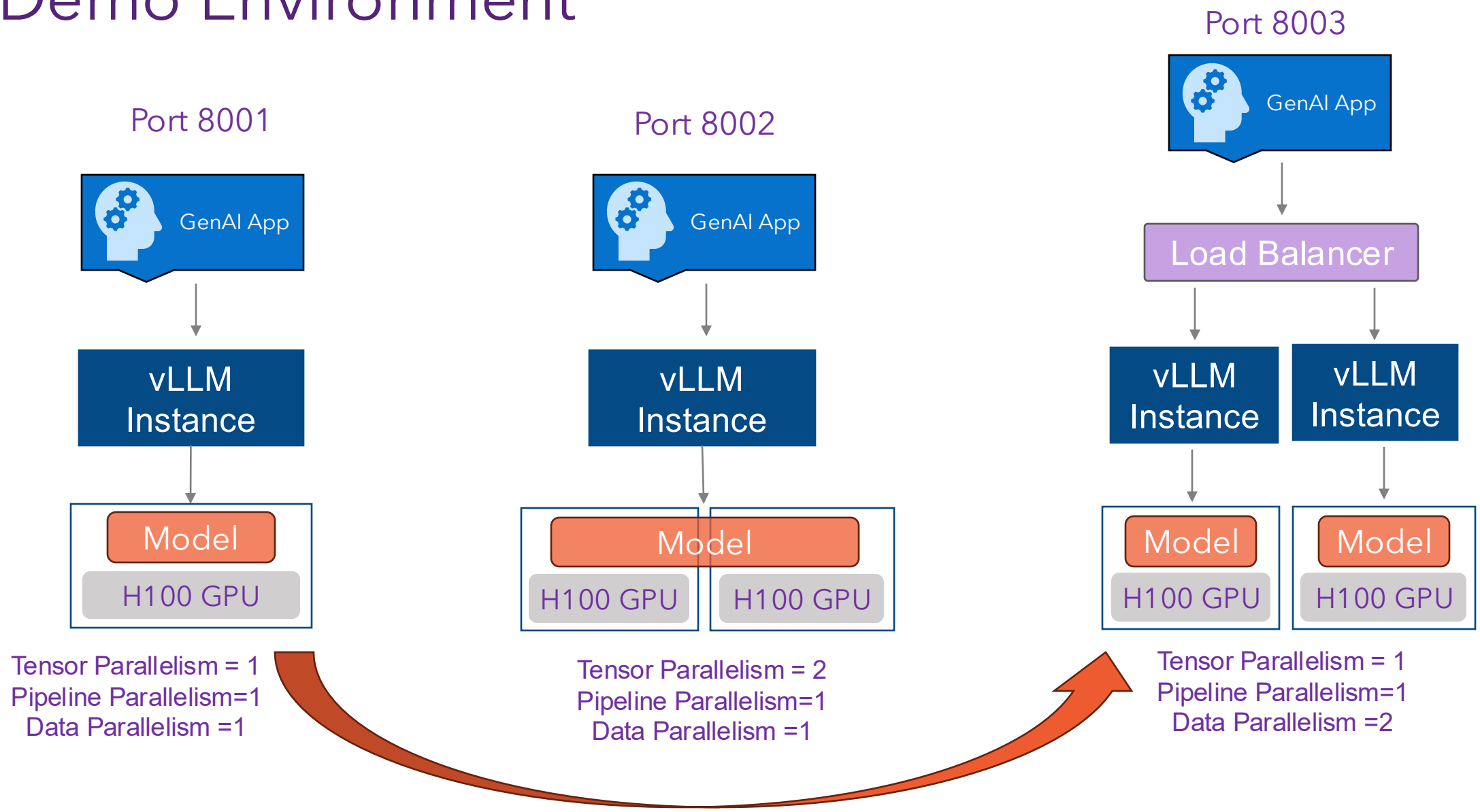




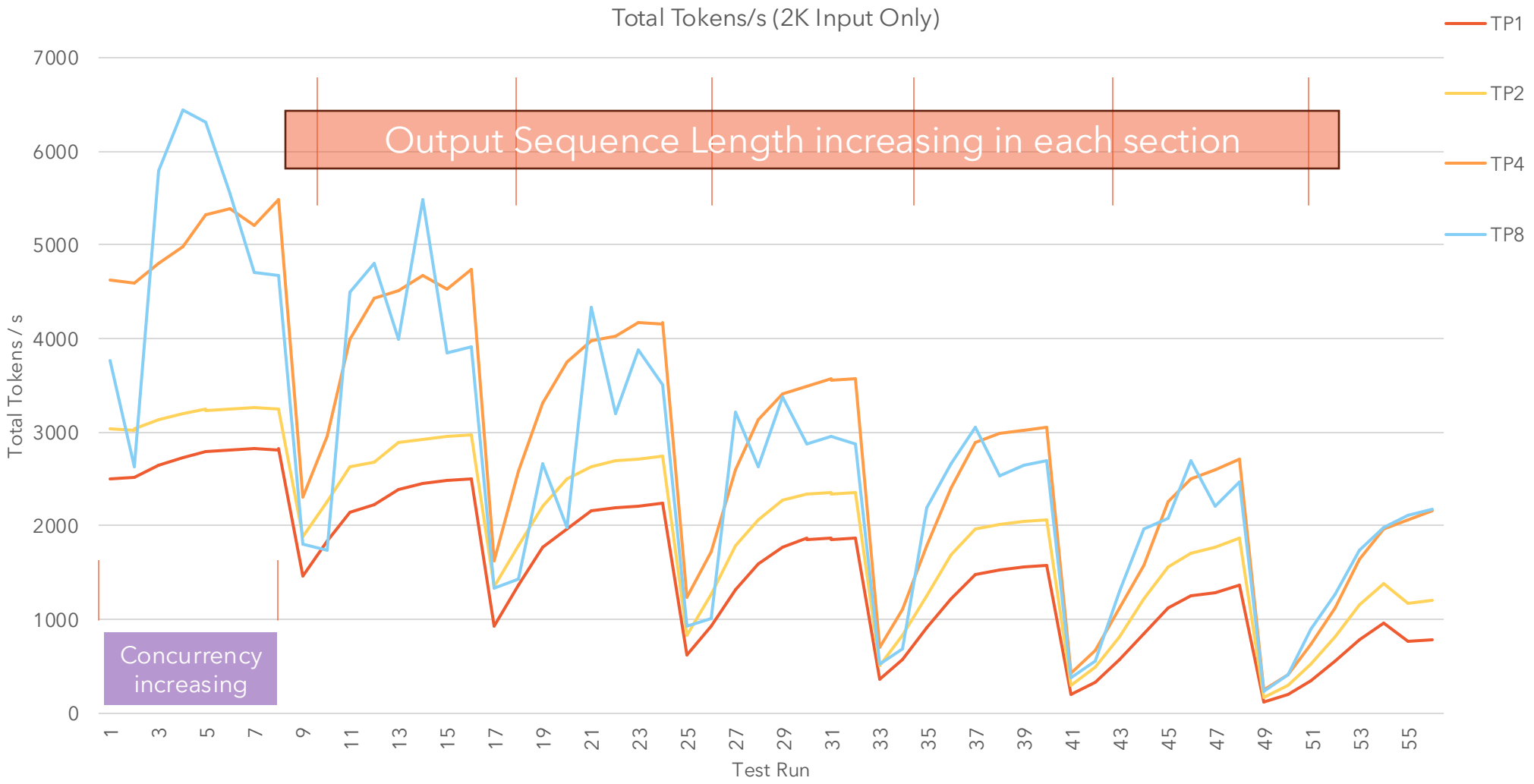
- Data Parallelism is running multiple copies of the model with some sort of load balancer/prompt router on top to scale the inferencing performance.
  - Entire Model replicated on each instance, reducing total space available for KV-cache vs other parallelism method.
- Tensor Parallelism is vertically splitting of the model among multiple GPUs.
  - Used when Model does not fit on a single GPU or looking for lower latency at low concurrencies.
  - Can be run in a multi-node configuration, but normally restricted to within a node because of higher bandwidth interconnects (e.g., NVLink) needed for Collectives (All-reduce and All-gather)
- Pipeline Parallelism is horizontally splitting the model between the layers.
  - Used when Model does not fit on a single GPU and have low bandwidth between GPUs (only PCIe) in a node or used in multi-node inferencing.
  - Need to deal with pipeline stalls (micro-batches)
- Parallelism techniques can be layered

Expert Parallelism (EP) and Wide-EP not covered.

Demo Environment

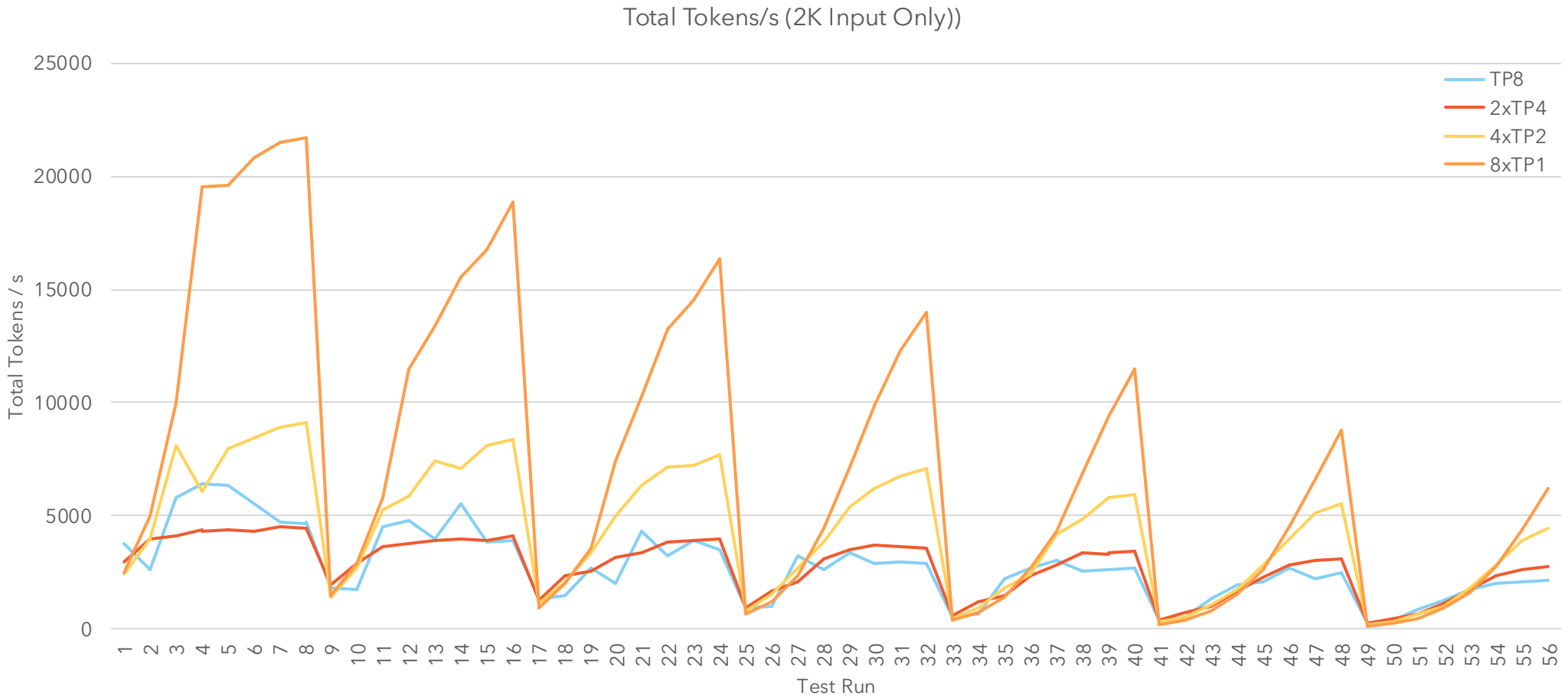


Tensor Parallelism (Example performance)



SNIA | DSN A SNIA Community

Data and Tensor Parallelism Combinations

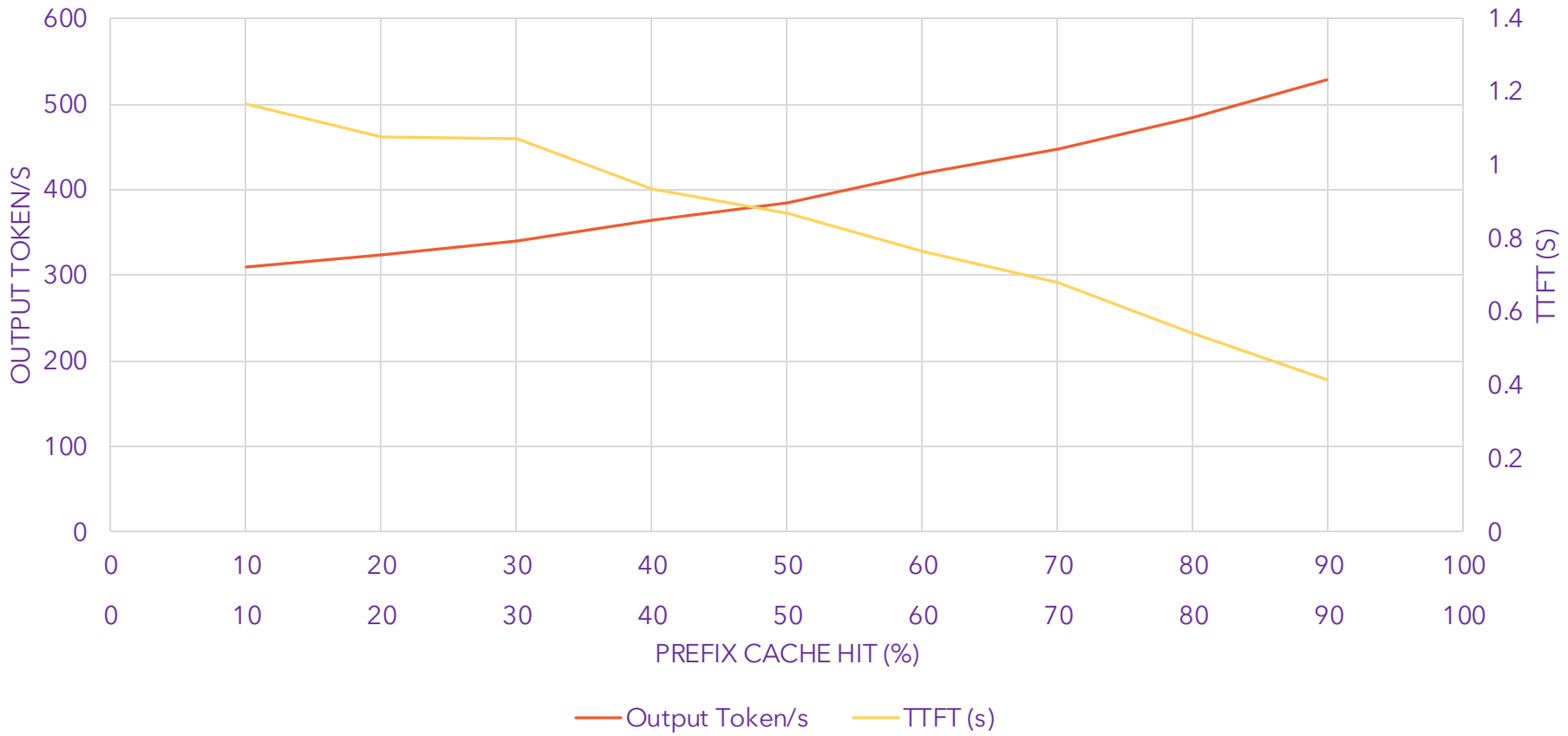


SNIA | DSN A SNIA Community

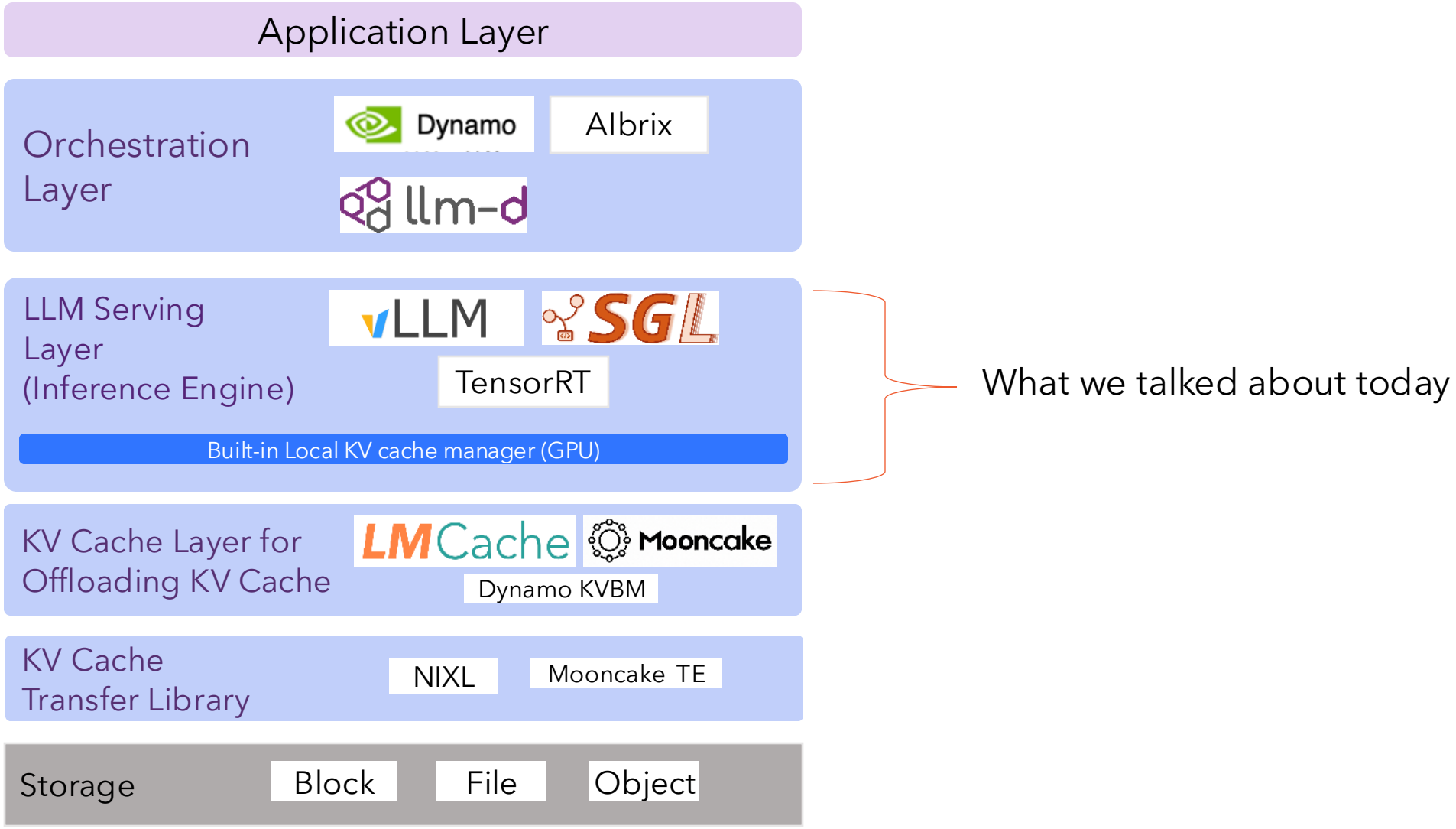
Prefix Caching

- Prefix Caching allows the reuse of the KV cache of prior requests, so that a new request can directly reuse the KV cache if it shares the same prefix with one of the prior requests, allowing the new request to skip the KV-cache computation of the shared part.
- Hashes of token blocks are used to determine commonality with existing KV cache blocks.
  - Cached KV blocks "A dog has a bone that …", new request "A dog has a bone which"
  - Red text is common and KV cache blocks representing that part can be reused.
- Now think of RAG/Conversational history, which would have a high degree of reuse, as the chat history is aggregated at every turn.

ISL/OSL/Concurrency 7000/500/20



AI Inference Stack: From Orchestration to Storage



SNIA SDC 25

SNIA | DSN
A SNIA Community

Q&A

After this Webinar

- Please rate this webinar and provide us with your feedback
- This webinar and a copy of the slides are available at the SNIA Educational Library snia.org/educational-library
- A Q&A from this webinar, including answers to questions we couldn't get to today, will be posted on our blog at sniablog.org
- Follow us on LinkedIn and X @SNIA

SNIA | DSN
A SNIA Community

Thank You