

# Orca: A Distributed Serving System for Transformer-Based Generative Models

Gyeong-In Yu and Joo Seong Jeong, Seoul National University; Geon-Woo Kim, FriendlyAI and Seoul National University; Soojeong Kim, FriendlyAI; Byung-Gon Chun, FriendlyAI and Seoul National University

OSDI 2022

# Generative Models

- Example of Summarization Task

## Prompt

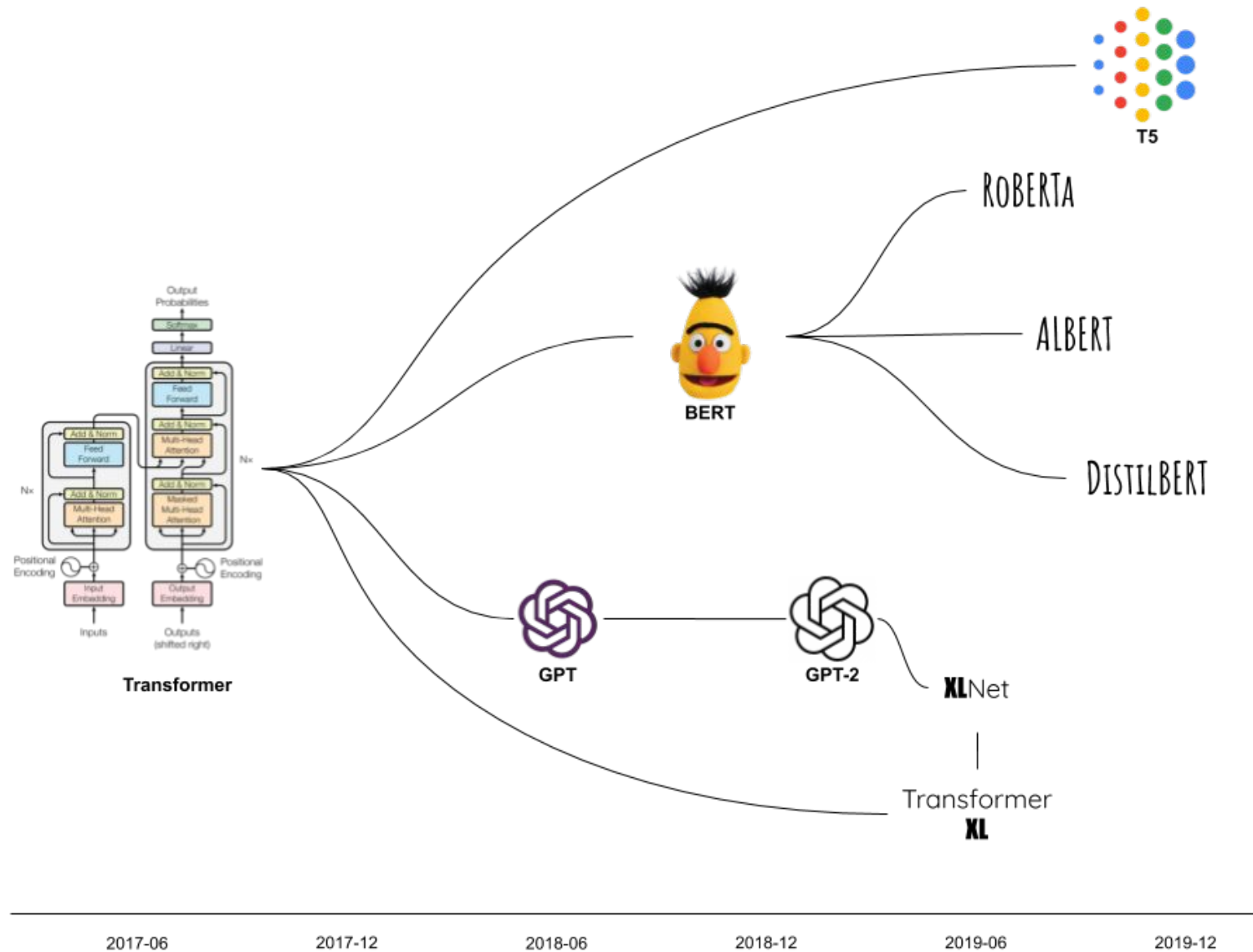
A neutron star is the collapsed core of a massive supergiant star, which had a total mass of between 10 and 25 solar masses, possibly more if the star was especially metal-rich.[1] Neutron stars are the smallest and densest stellar objects, excluding black holes and hypothetical white holes, quark stars, and strange stars.[2] Neutron stars have a radius on the order of 10 kilometres (6.2 mi) and a mass of about 1.4 solar masses.[3] They result from the supernova explosion of a massive star, combined with gravitational collapse, that compresses the core past white dwarf star density to that of atomic nuclei.

Tl;dr

## Sample response

A neutron star is the collapsed core of a massive supergiant star. These ultra-dense objects are incredibly fascinating due to their strange properties and their potential for phenomena such as extreme gravitational forces and a strong magnetic field.

# Era of Generative Models



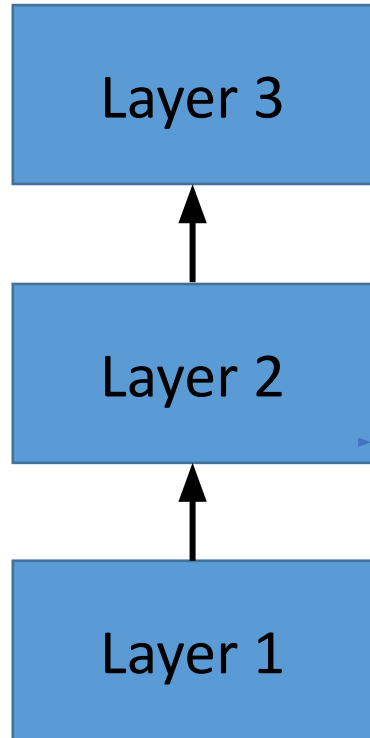
# Cost of Serving

- In Azure, a GPT3 175B instance requires 2 VMs, each of which has 8 NVIDIA A100 40GB GPUs
- At Azure US East, the VM price is \$21.197/hour

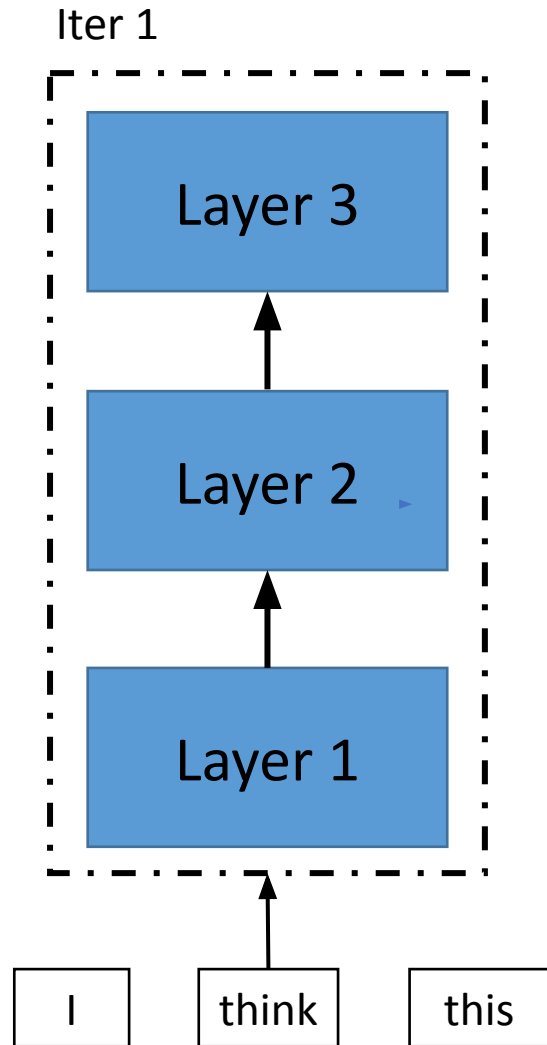
So, the yearly price for hosting 400 GPT3 175B instance is  
~190.6Million/year

**This paper focuses on how to improve the  
throughput of serving transformer-based  
generative models to reduce the cost**

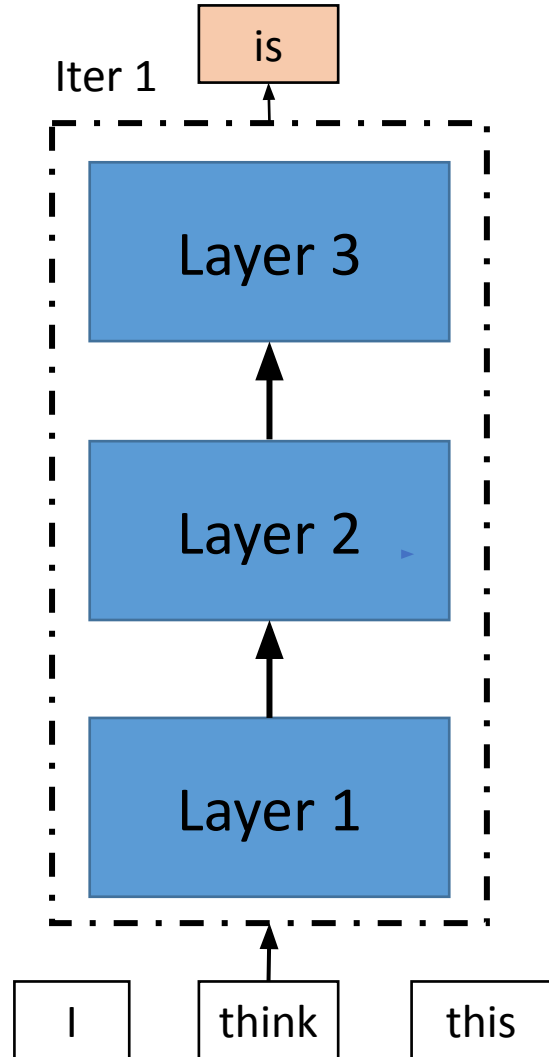
# Inference of Generative Models



# Inference of Generative Models

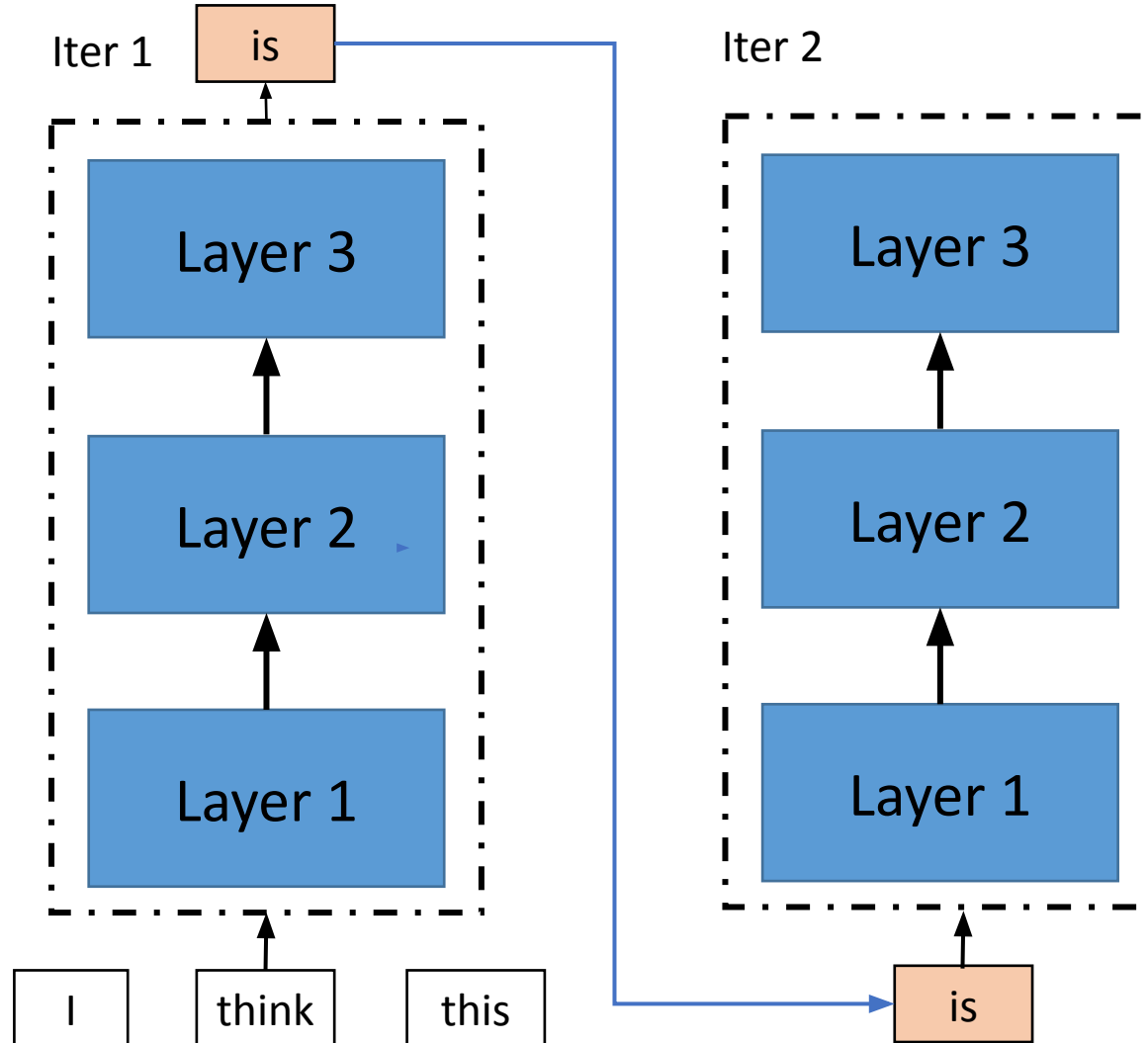


# Inference of Generative Models

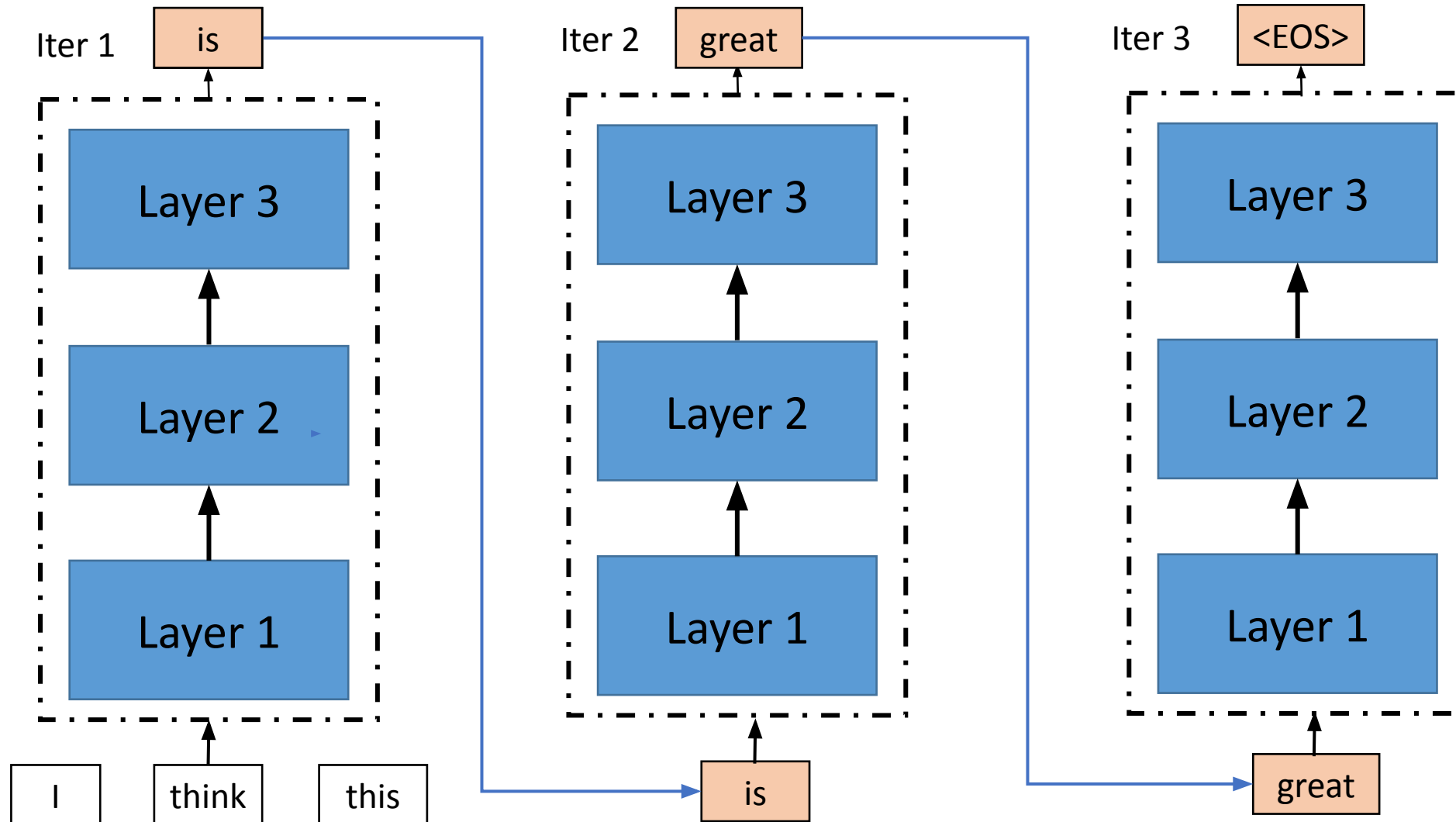




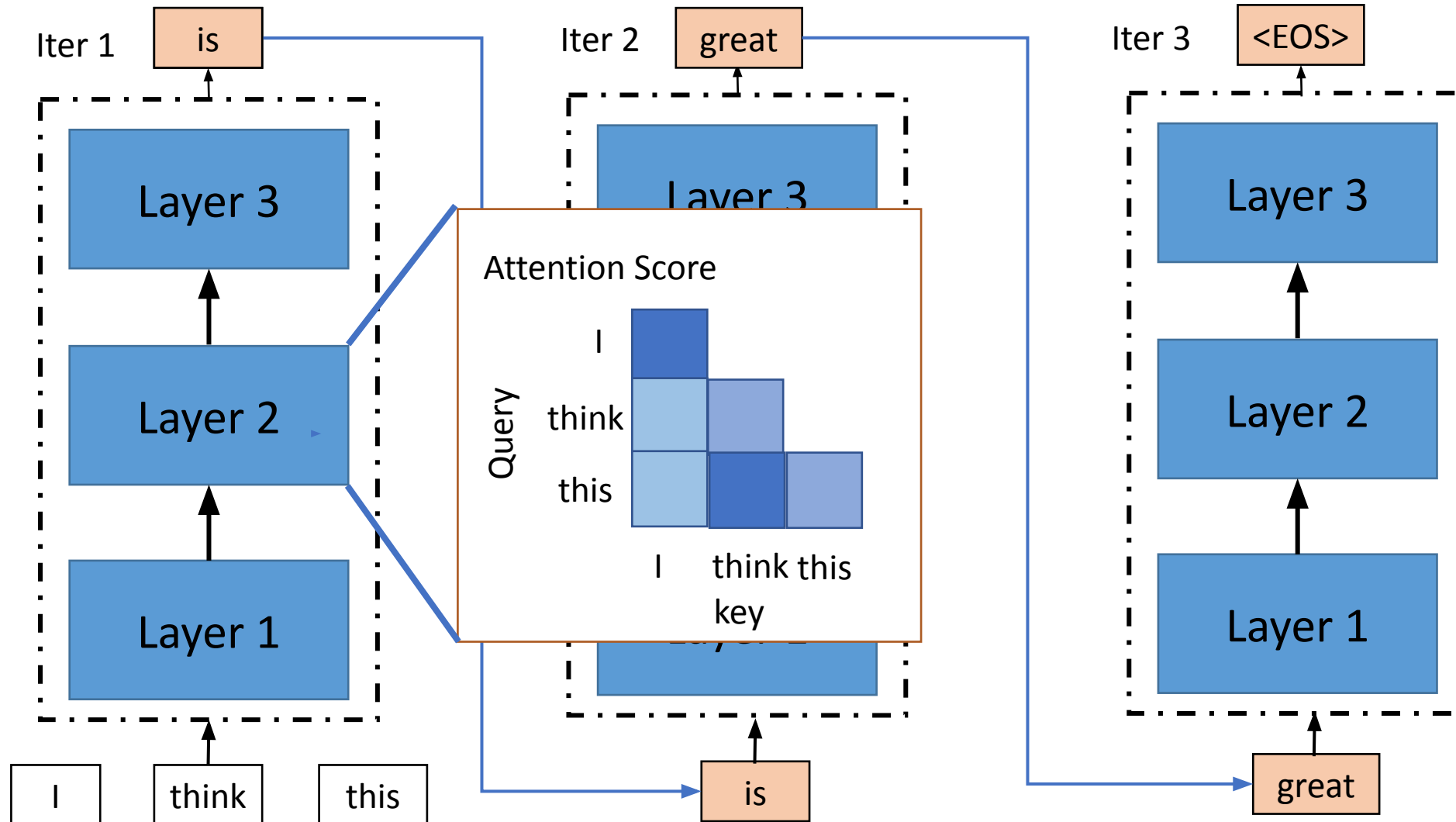
# Inference of Generative Models



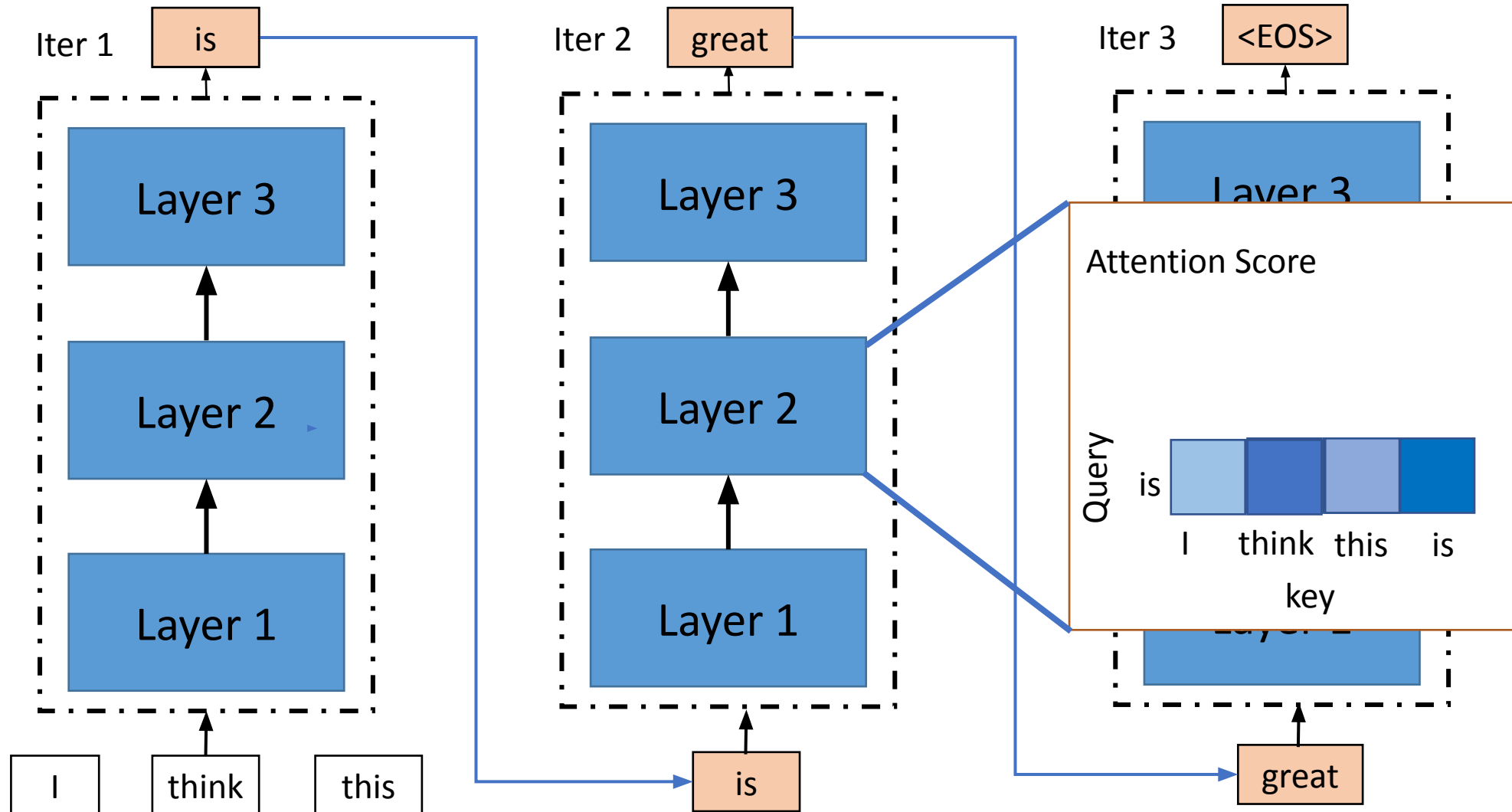
# Inference of Generative Models



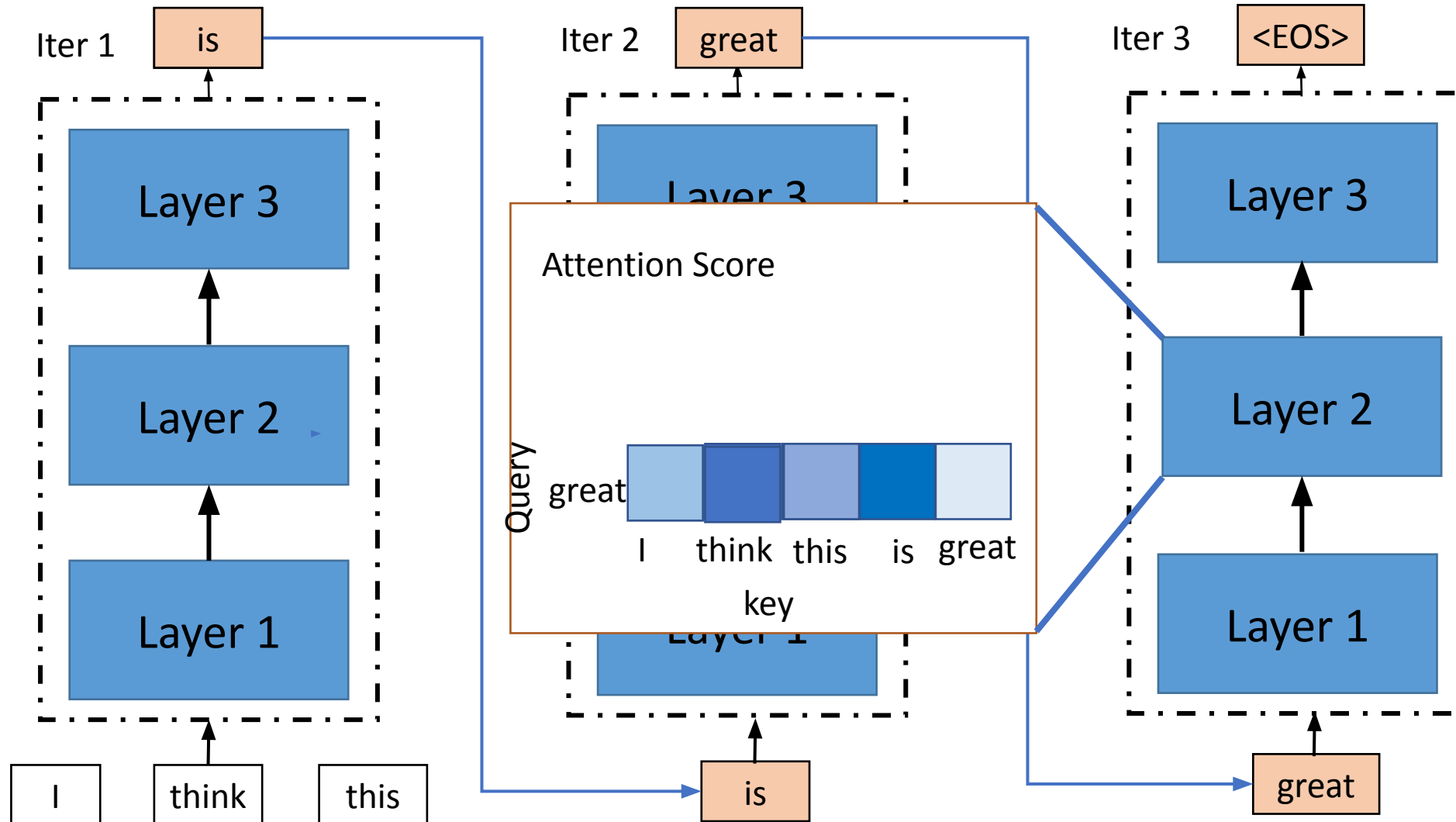
# Inference of Generative Models



# Inference of Generative Models



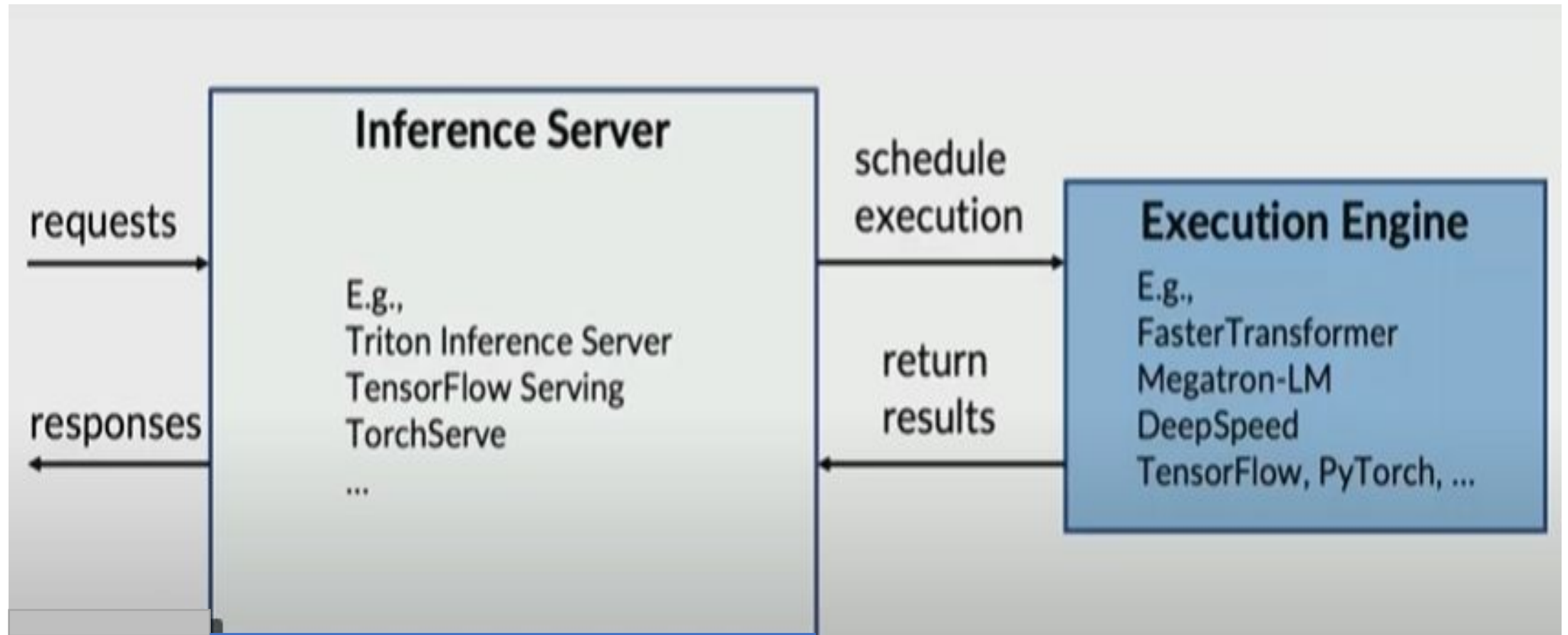
# Inference of Generative Models



# Characteristics of Inference of Generative Models

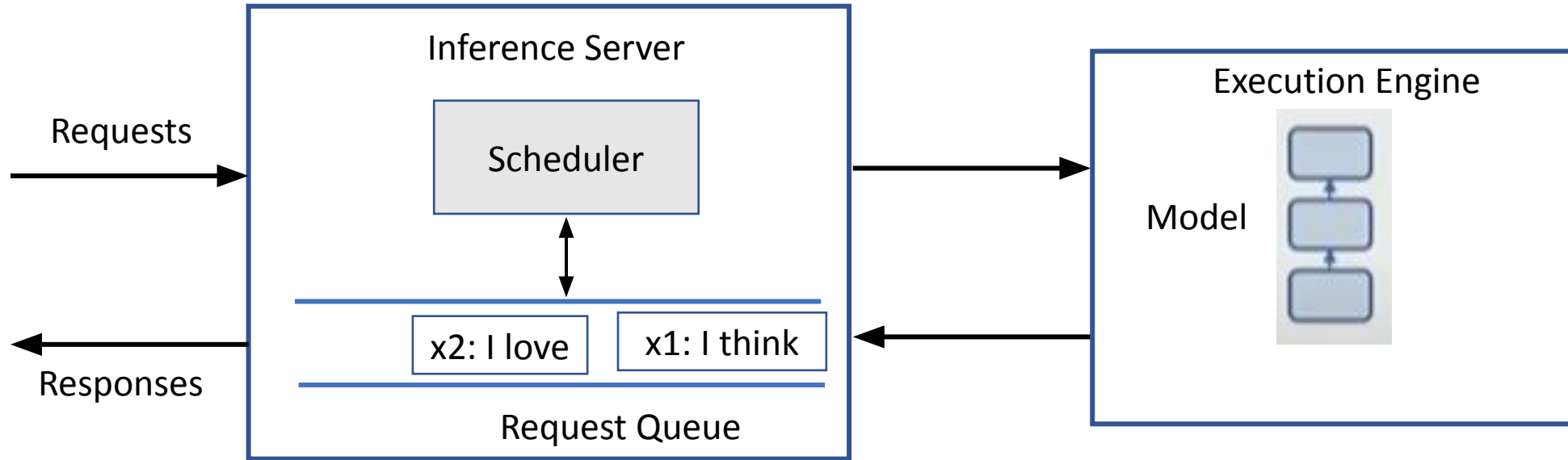
- Multi iteration characteristic
  - Generate one token at a time
- Initiation phase (1<sup>st</sup> iteration)
  - Process all input tokens at once
- Increment phase (2<sup>nd</sup> – last iterations)
  - Process a single token generated from the previous iteration
  - Use Attention keys and values of all previous tokens
- Save Attention keys and values for the following iterations to avoid recomputation

# Serving of Generative Language Models



# Serving of Generative Language Models

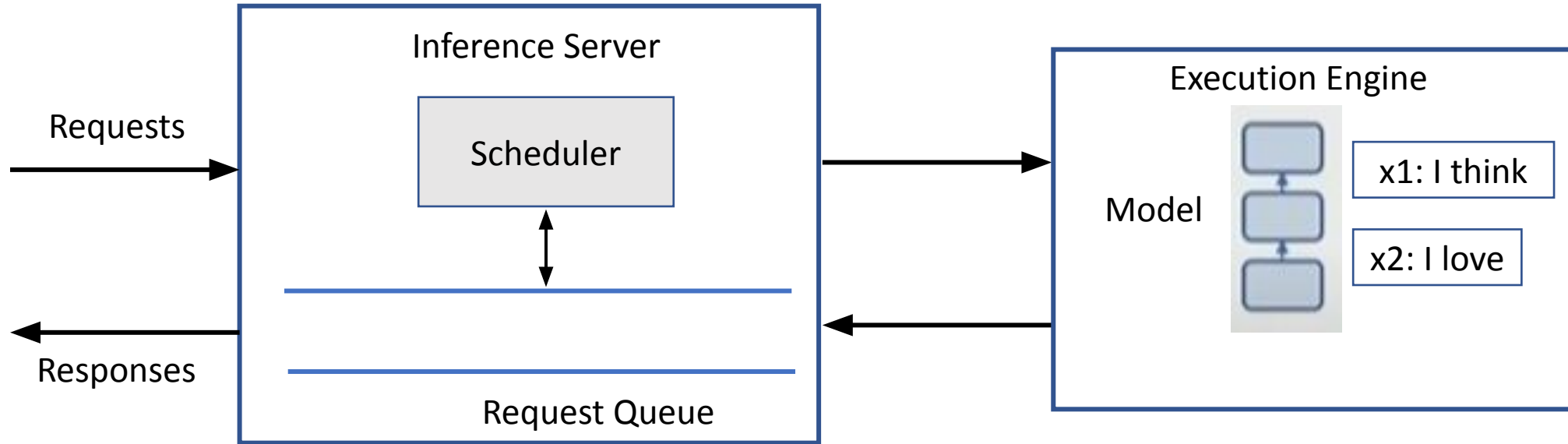
Maximum batch size = 3





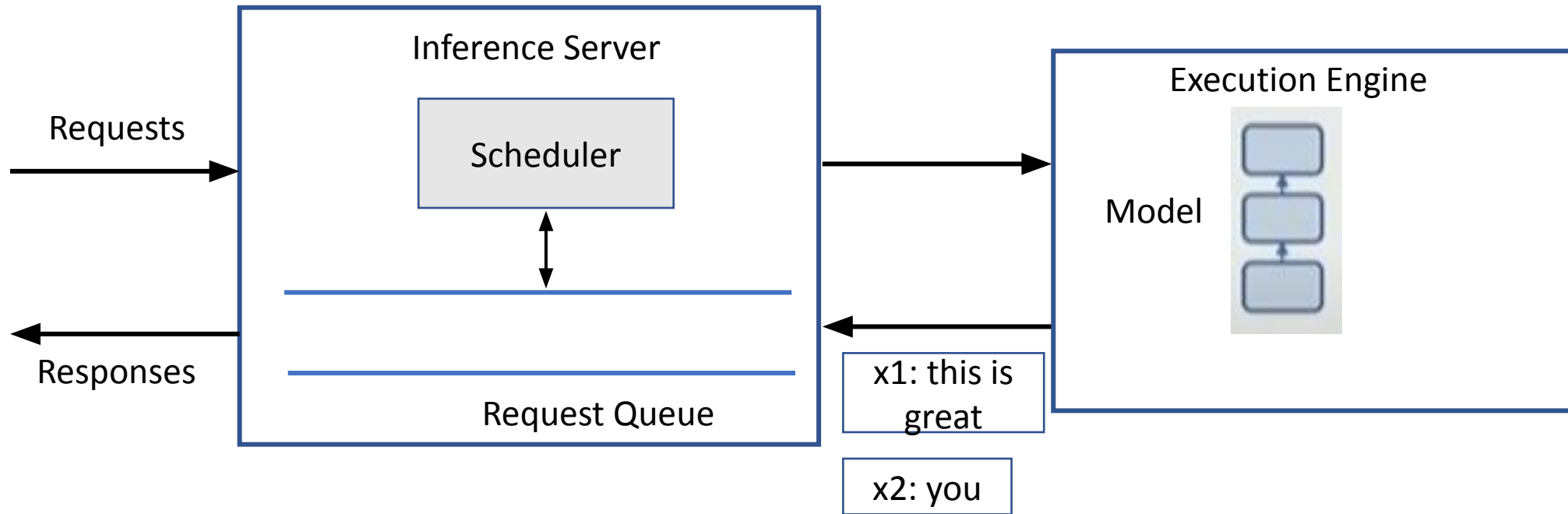
# Serving of Generative Language Models

Maximum batch size = 3



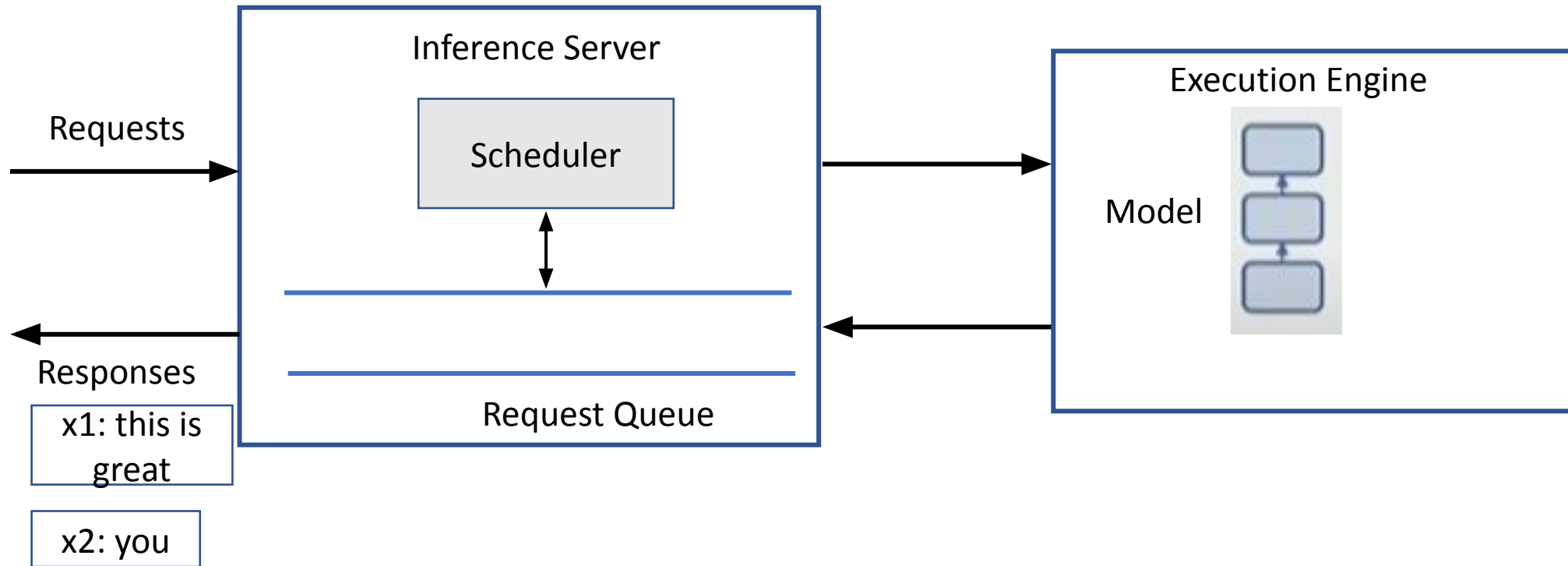
# Serving of Generative Language Models

Maximum batch size = 3



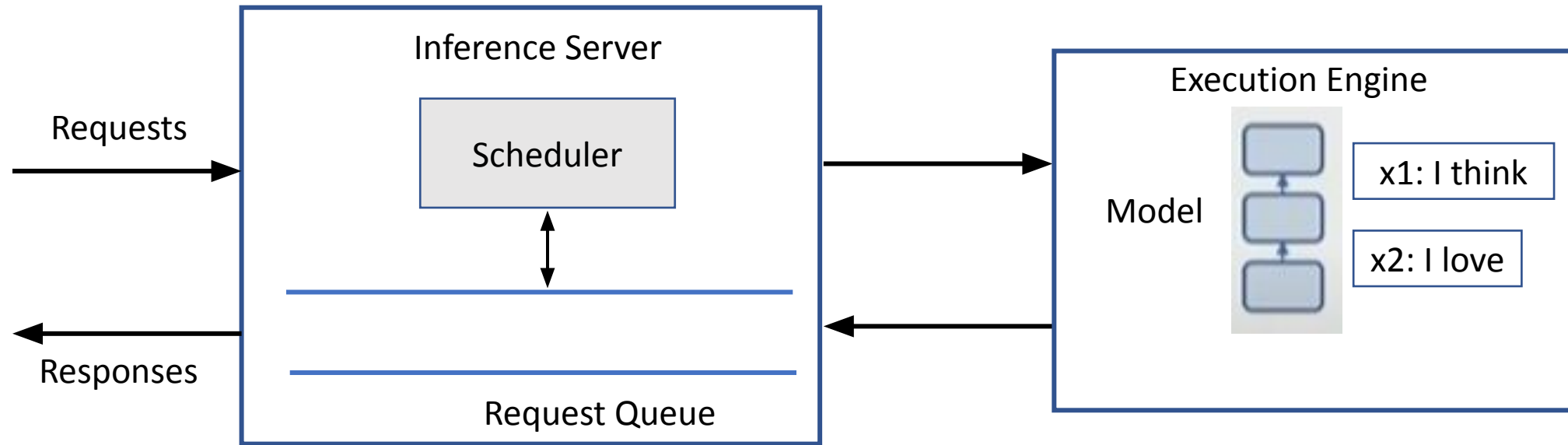
# Serving of Generative Language Models

Maximum batch size = 3



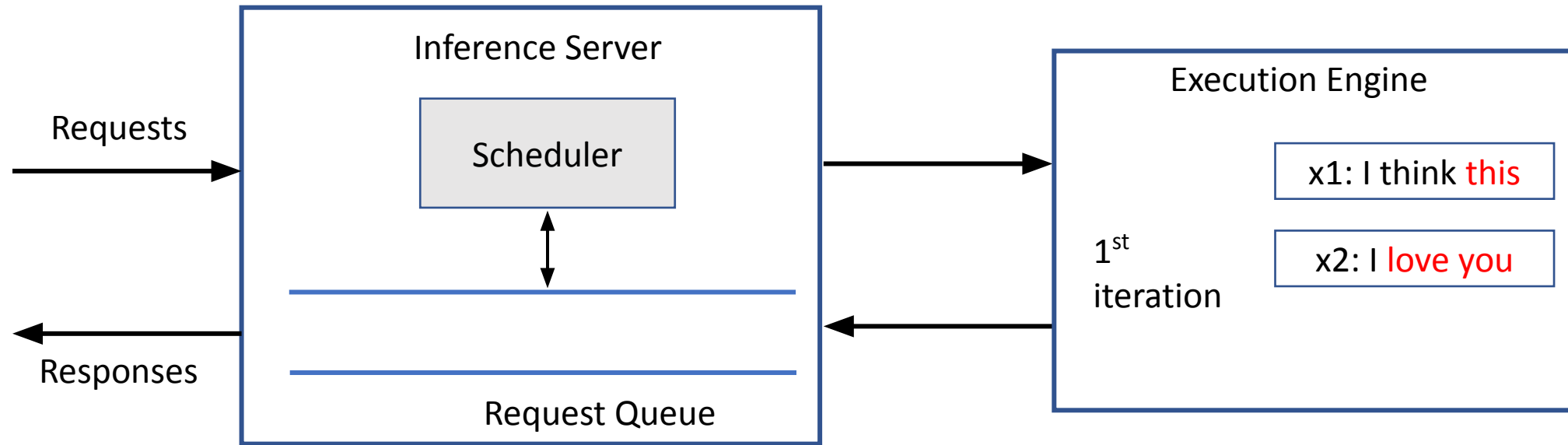
# Problem 1: Request Level Scheduling

Maximum batch size = 3



# Problem 1: Request Level Scheduling

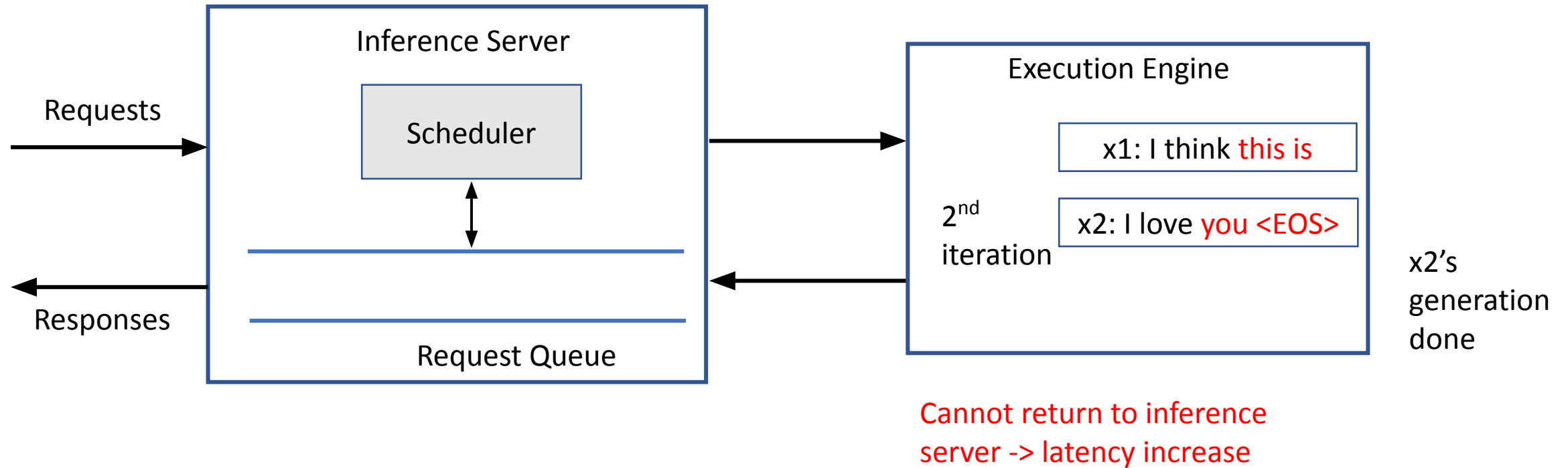
Maximum batch size = 3



Process the requests until they are done

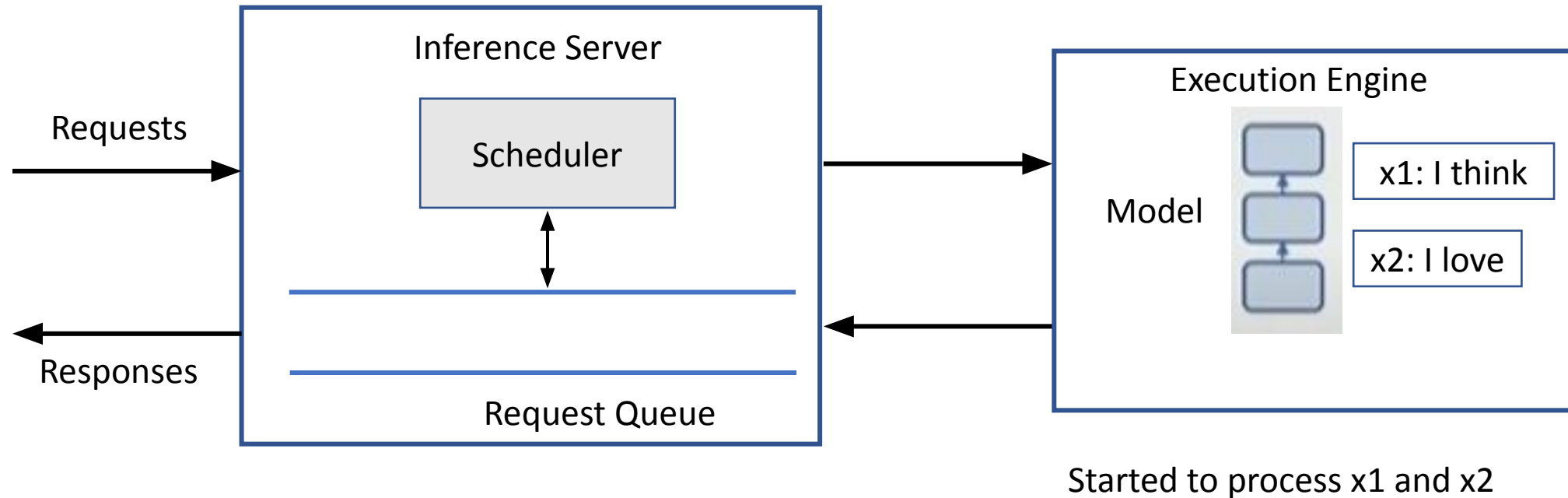
# Problem 1: Request Level Scheduling

Maximum batch size = 3



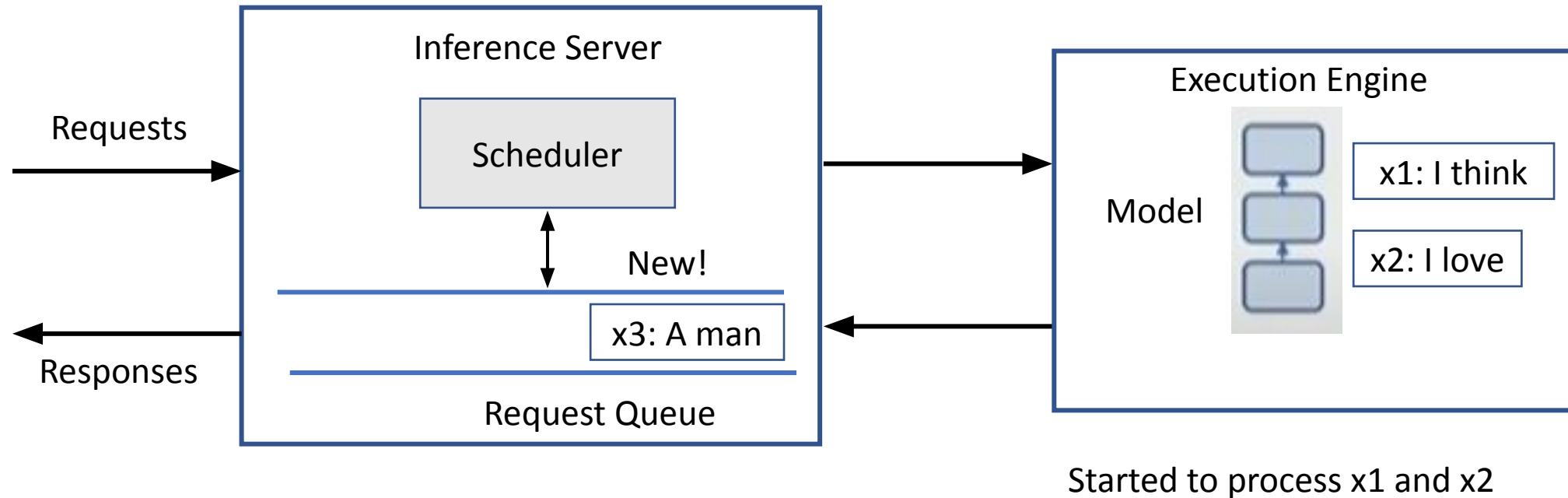
# Problem 1: Request Level Scheduling

Maximum batch size = 3



# Problem 1: Request Level Scheduling

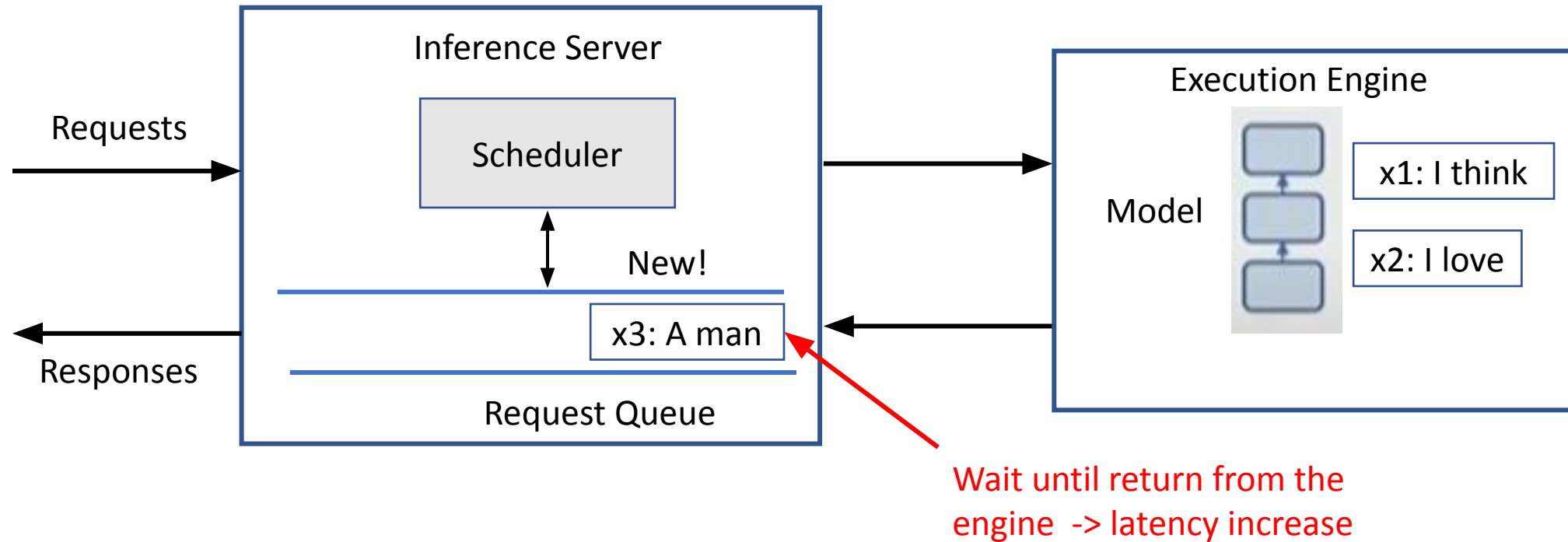
Maximum batch size = 3





# Problem 1: Request Level Scheduling

Maximum batch size = 3



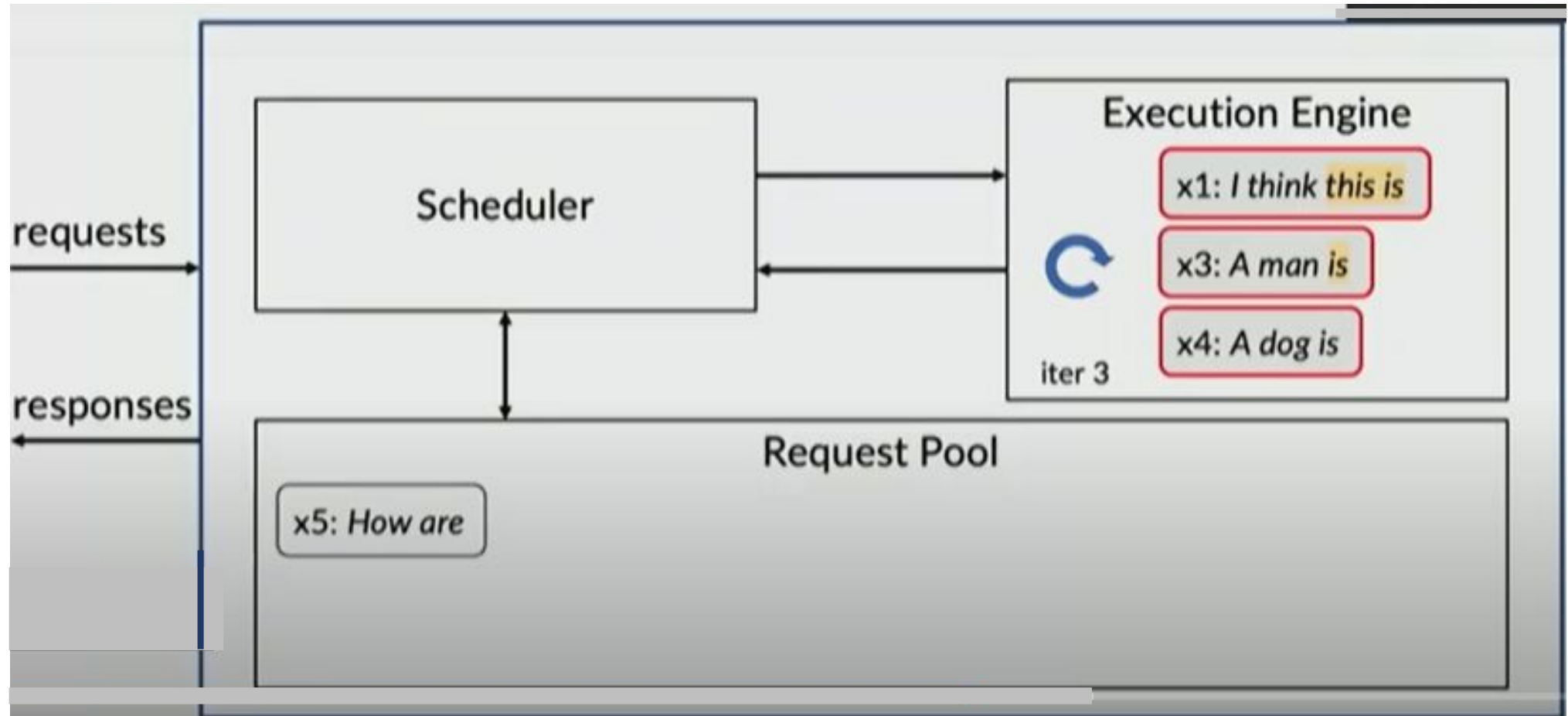
# Solution 1

Iteration-Level Scheduling

# Iteration Level Scheduling

Can handle early-finished or late-arrived requests more efficiently

## Problem 2: Batching



# Batching Requirements

- There are three cases for a pair of requests where the next iteration cannot be batched together:

(1) both requests are in the initiation phase and each has different number of input tokens

(2) both are in the increment phase and each is processing a token at different index from each other

or

(3) each request is in the different phase: initiation or increment

# Batching Requirements

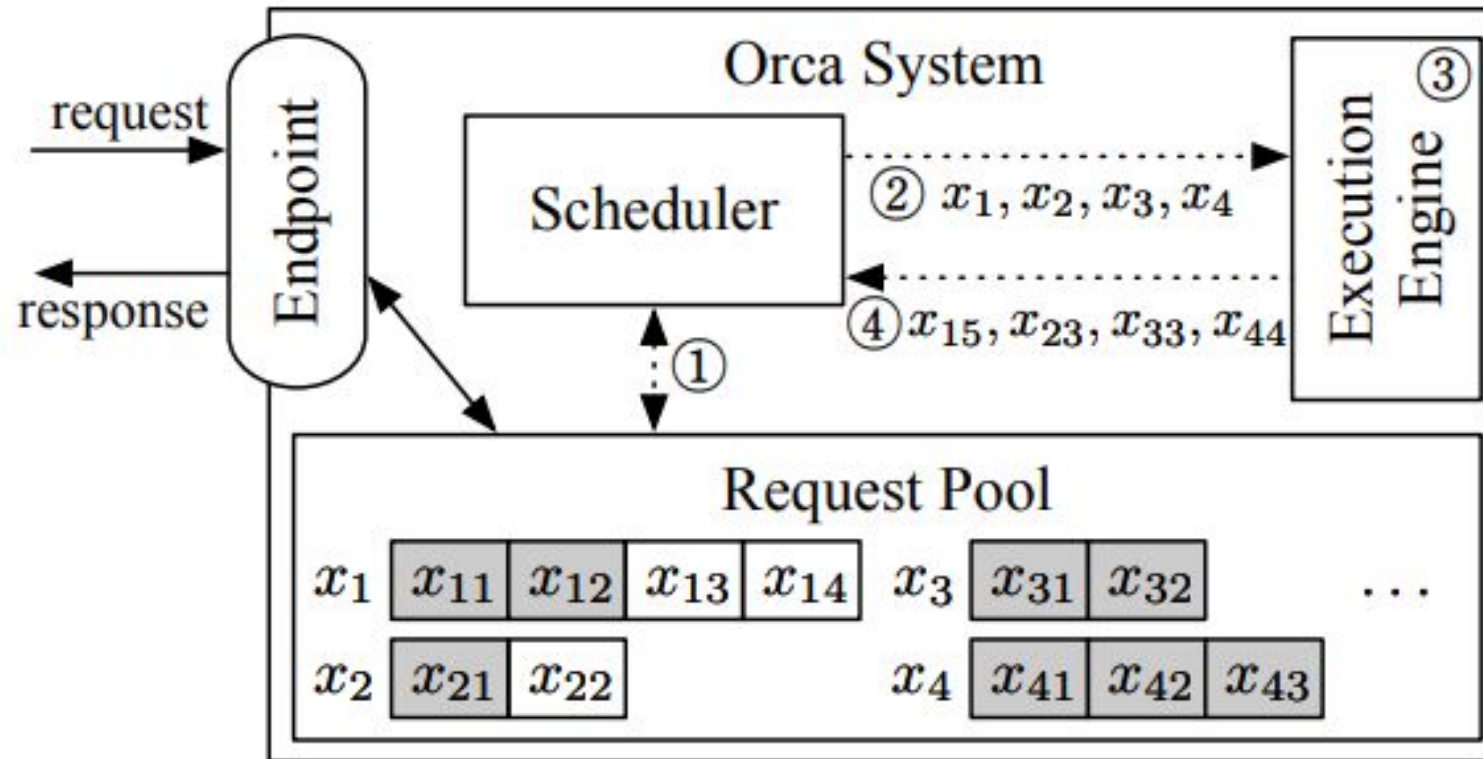
- That is batching is only applicable
  - - Requests are at same phase
  - - Requests are of same length

# Solution 2: Selective Batching

## Solution 2

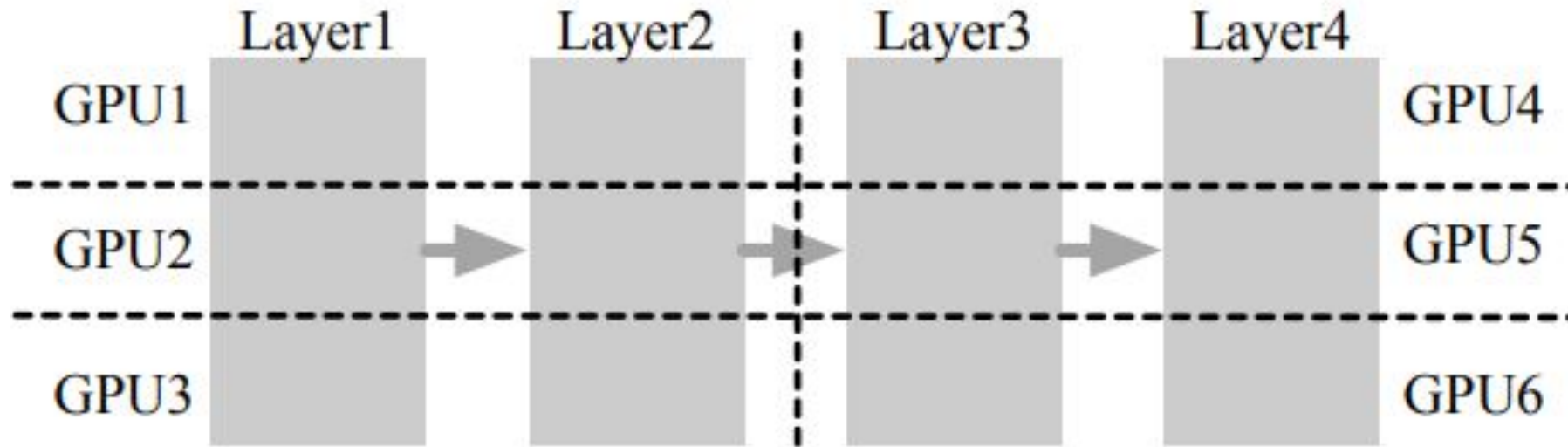
Selective Batching

# ORCA System Architecture

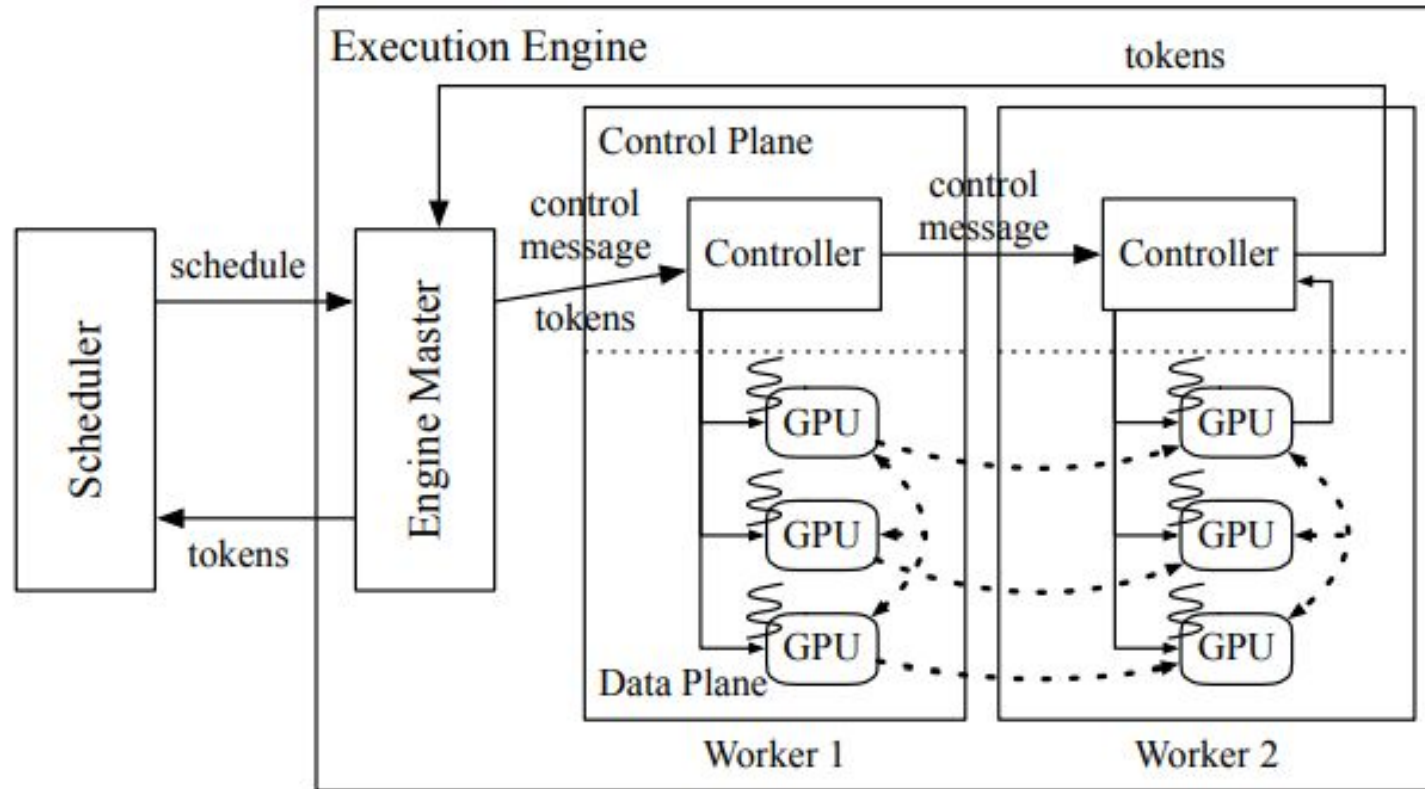




# ORCA System Architecture



# ORCA System Architecture



# Scheduling

- Simple first-come-first-served algorithm
- Efficient pipelining across multiple workers
- Memory management for saving the Attention keys and values
- Orca configures the maximum batch size knob

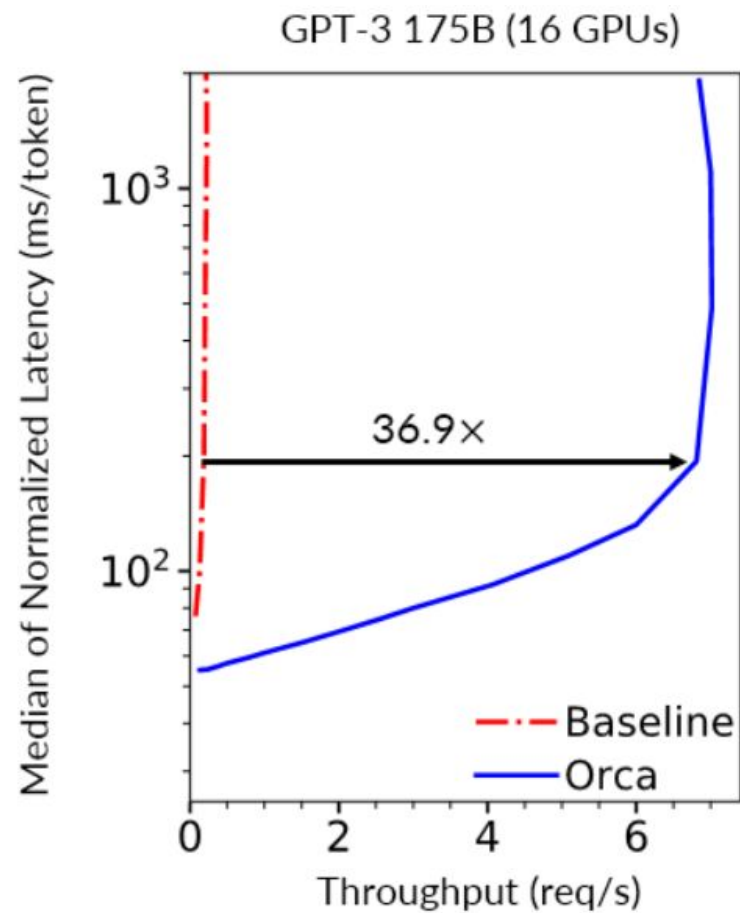
# Evaluation Setup

- Model
  - GPT-3 models upto 341B parameters
- Hardware Setup
  - Azure ND96asr A100 v4 VMs, each equipped with 8 NVIDIA 40-GB A100 GPUs
  - Each VM has 8 Mellanox 200Gbps HDR Infiniband adapters
- Baseline
  - Execution Engine: NVIDIA FasterTransformer
  - Inference Server: custom scheduler that mimics the batch scheduler of the Triton inference server

# Evaluation Setup

- Workload
  - Synthesized the trace of client requests
  - Request arrival time: Poisson process with varying request rate
  - Input length: Uniform(32,512)
  - Output length: Uniform(1,128)
- Metric – throughput-latency

# Results



# Cost of Serving

- The yearly price for hosting 400 GPT3 175B instance is

• ~190.6 Million/year

Baseline



~5.7 Million/year

Orca

# Conclusion

- Orca is the first serving system for **Transformer-based** generative models that employs **iterative scheduling** and **selective batching**
- Orca improves the **throughput** of GPT-3 175B by up to **36.9X** for the **same level of latency**
- Orca is currently deployed in FriendlyAI's cloud service