CSCE 585: Machine Learning Systems

Lecture 4: Designing and Motivating (ML) Systems Experiments: Lessons from InferLine

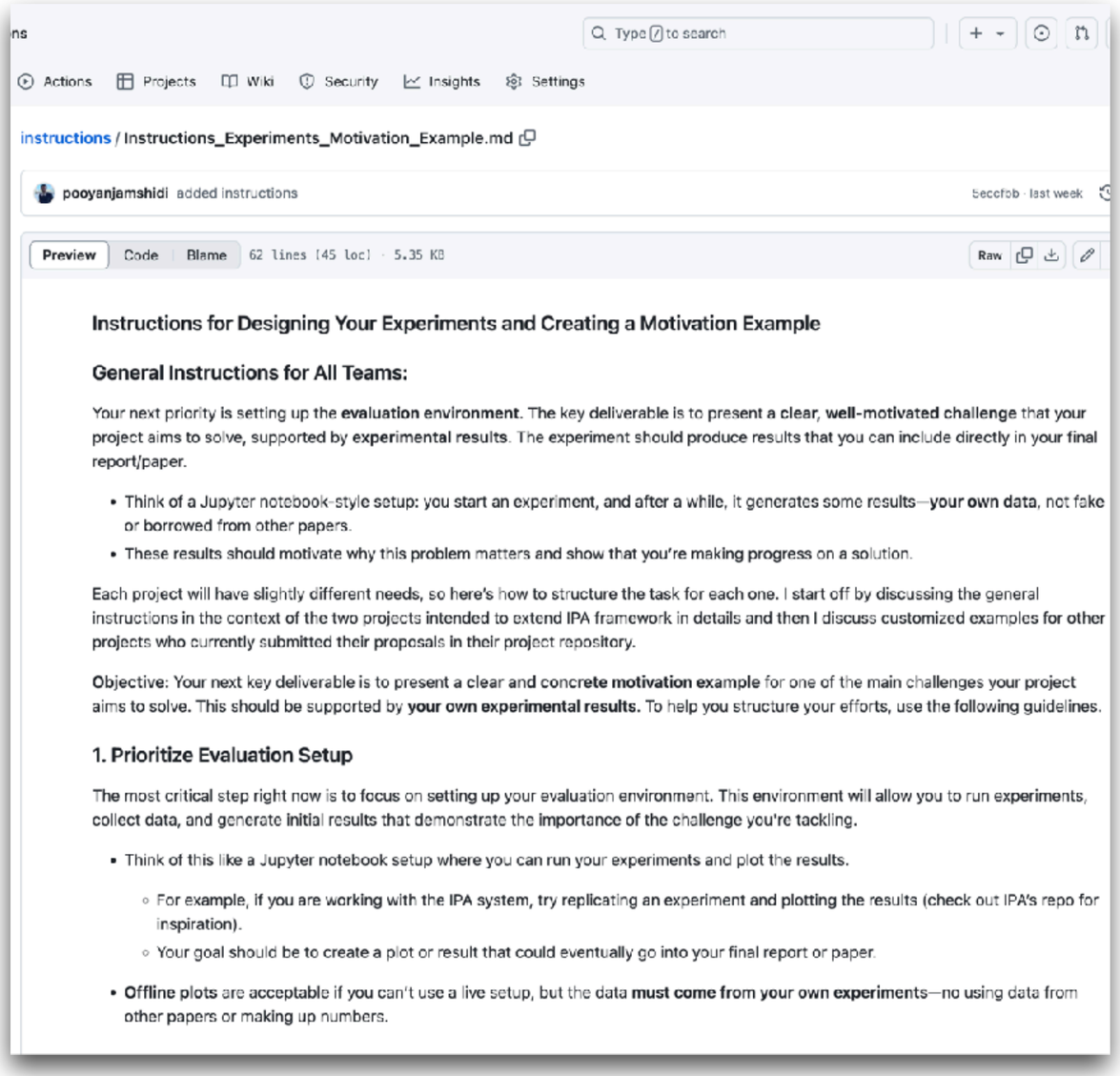
Pooyan Jamshidi

Fall 2024 | Course Website: https://pooyanjamshidi.github.io/mls/

UNIVERSITY OF South Carolina

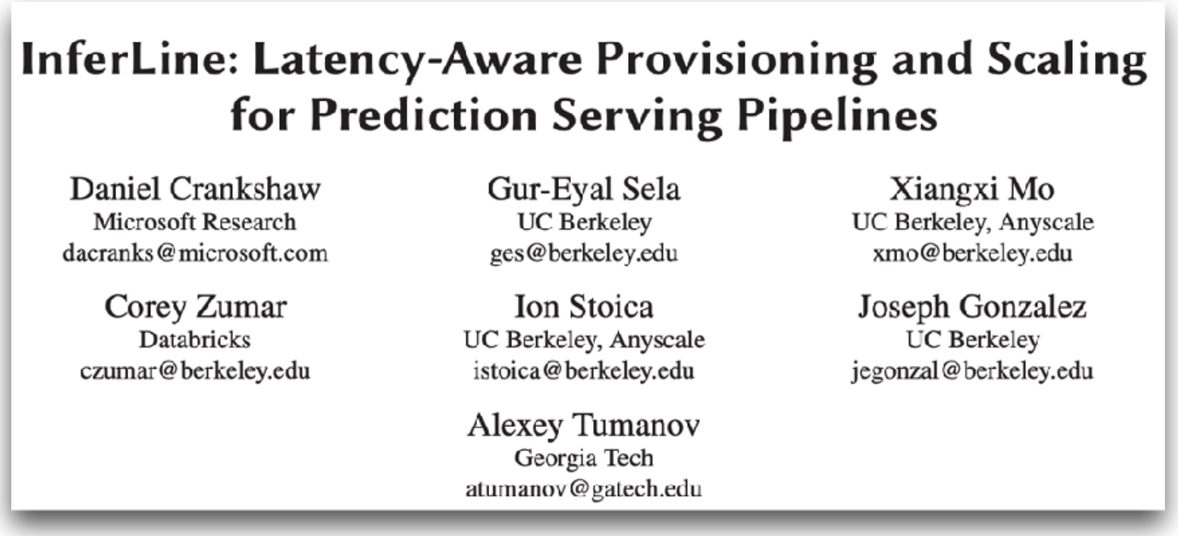
Previously!

We have discussed detailed instructions on preparing the motivation example for your project.



In this lecture

Now, we discuss the approach in the context of a MLSys paper we have recently read—i.e., InferLine.



InferLine: Latency-Aware Provisioning and Scaling for Prediction Serving Pipelines

Daniel Crankshaw
Microsoft Research
dacranks@microsoft.com

Gur-Eyal Sela
UC Berkeley
ges@berkeley.edu

Xiangxi Mo
UC Berkeley, Anyscale
xmo@berkeley.edu

Corey Zumar
Databricks
czumar@berkeley.edu

Ion Stoica
UC Berkeley, Anyscale
istoica@berkeley.edu

Joseph Gonzalez
UC Berkeley
jegonzal@berkeley.edu

Alexey Tumanov
Georgia Tech
atumanov@gatech.edu

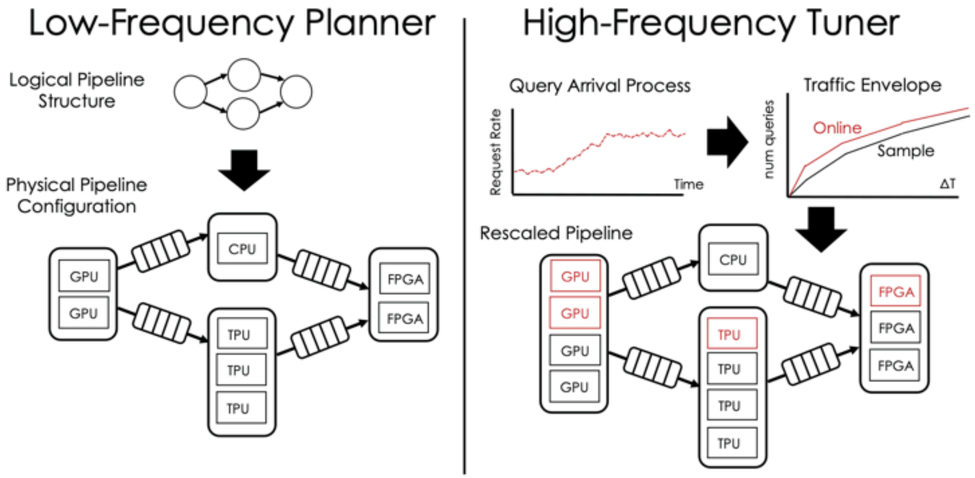
Objective of the Experiment

Motivating Project Experiments: A Case Study with InferLine

- **Goal**: Show how to experimentally motivate a project by running experiments and producing results that directly address a challenge.
- **InferLine Focus**: Latency-Aware Provisioning and Scaling for Prediction Pipelines.
- **Student Action**: Understand the relevance of your project's core challenge by setting up experiments to gather data.

Overview of InferLine

- **Problem**: Managing and provisioning machine learning inference pipelines with strict tail latency Service Level Objectives (SLOs).
- **Challenges**:
  - Hardware heterogeneity (CPU, GPU, TPU).
  - Combinatorial configuration space.
  - Handling bursty workloads while minimizing cost.
- **Key Contributions**: Two components:
  - Low-frequency Planner: Optimizes pipeline configuration periodically.
  - High-frequency Tuner: Reacts to workload changes rapidly.



Designing Experiments – Key Components

Designing a Motivation Experiment Using InferLine as an Example

- **Independent Variables**: Model size, hardware type, and batch size.
- **Dependent Variables**: Latency, cost, and query throughput.
- **Control Variables**: Fixed models and hardware resources.
- **Example**:
  - Running InferLine on varying hardware configurations (GPU, CPU).
  - Measuring the change in end-to-end latency and cost.

Experimental Setup – InferLine

Setting Up Experiments Based on InferLine

- **Environment Setup**: Use a simulation or real inference pipeline environment (TensorFlow Serving or Clipper).
- **Run Scenarios**:
  - Different batch sizes.
  - Test under bursty workloads with various model and hardware configurations.
- **Metrics:**
  - Throughput (queries per second).
  - P99 latency (99% of queries below the given latency threshold).
  - Cost efficiency (CPU vs GPU usage).

Motivating with Results

InferLine Results: Motivation for Addressing Latency Constraints

- **Example Plot**: Show how increasing batch size impacts latency and cost.
- **Plot Example:** Latency (ms) vs. Batch Size for different hardware (GPU, TPU).
- **Plot Explanation:** As batch size increases, throughput improves, but latency also grows —leading to trade-offs.
- **Conclusion:** These results show the complexity of managing inference pipelines, motivating the need for an automated solution like InferLine.

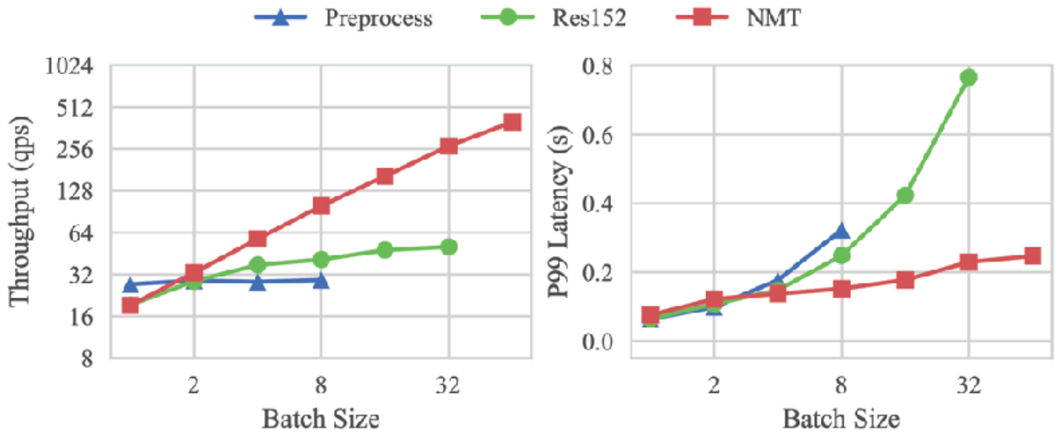


Figure 3: Example Model Profiles on K80 GPU. The preprocess model has no internal parallelism and cannot utilize a GPU. Thus, it sees no benefit from batching. Res152 (image classification) & TF-NMT(text translation model) benefit from batching on a GPU but at the cost of increased latency.

Takeaways for Your Project

Applying InferLine's Approach to Your Project

- Identify the key challenge (e.g., latency, cost, resource efficiency).
- Design an experiment that reflects that challenge.
- Generate real data that helps motivate your project’s problem and solutions.

Next Steps

- Set up your **evaluation environment**
- Run initial **experiments**
- Produce **plots** that motivate the challenge.