Background and Motivation

1. Why a dedicated system for ML Inference?
   ML serving has many distinct properties:
   - compute intensive;
   - deterministic processing time;
   - can benefit from hardware accelerators, e.g., GPUs, TPUs;
   - stateless.

2. Design objectives
   - serve ML inference on public cloud;
   - scale quickly to dynamic queries;
   - cost-effective;
   - SLO-aware.

Challenges:
- Which cloud services to use, Infrastructure as a Service (IaaS), Container as a Service (CaaS), or Function as a Service (FaaS)?
- How to navigate through the large configuration space?
- How to leverage cost-performance tradeoffs such as preemptable instances and burstable instances?

2. Design objectives
   - serve ML inference on public cloud;
   - scale quickly to dynamic queries;
   - cost-effective;
   - SLO-aware.

Challenges:
- Which cloud services to use, Infrastructure as a Service (IaaS), Container as a Service (CaaS), or Function as a Service (FaaS)?
- How to navigate through the large configuration space?
- How to leverage cost-performance tradeoffs such as preemptable instances and burstable instances?

Characterization Highlights

4. IaaS vs CaaS vs FaaS
   - IaaS has the lowest cost and latency, but new instances takes minutes to launch.
   - FaaS scales well, but it is expensive with long latency.
   - CaaS offers the middle ground.

   ★ Combine the cost-effective IaaS and the scalable FaaS.

5. CPU vs GPU vs TPU
   - High utilization is required to justify the high cost of GPUs and TPUs
   - Judicious batching is needed
   - GPUs can be cheaper than CPUs
   - TPUs are not suitable for inference

   ★ Predict workload to maintain high utilization
   ★ Judicious batching to navigate the tradeoff between cost and latency

System Design

6. MArk (Model Ark) system design
   - uses IaaS (EC2) as the primary means of service;
   - employs FaaS (Lambda) to cover load spikes;
   - uses proactive provisioning, planning instances based on prediction;
   - dynamically batches requests according to instance types and request arrivals;
   - tracks SLO compliance, and launches burstable instances when needed.

Evaluation Highlights

7. What about performance?
   Cost savings:

   [Graphs and tables showing cost savings for various models and configurations]

   - Up to 3.6x cost reduction with only on-demand instances.
   - Up to 7.8x cost reduction if spot instances are considered.

Mark is open sourced at: https://github.com/marcoszh/MArk-Project