

# Scalable and Cost-Efficient ML Inference: Parallel Batch Processing with Serverless Functions


Amine Barrak and Emna Ksontini

Oakland University, Michigan, USA

## Motivation

**ML inference** is the process of using a trained model to make predictions on new data, typically by splitting the dataset into **batches** for either **monolithic (sequential)** or **parallel processing**.

Approach	Monolithic	Parallel
Description	The system processes one batch, completes its forward pass through the model, and then moves to the next batch (sequential)	Each computational unit (e.g., GPU or CPU thread) processes its assigned batch independently. The results from all batches are gathered
Pros	Low Cost	Fast Execution
Cons	Latency	High Cost
<i>Ressource under-provisioning or over-provisioning</i>		

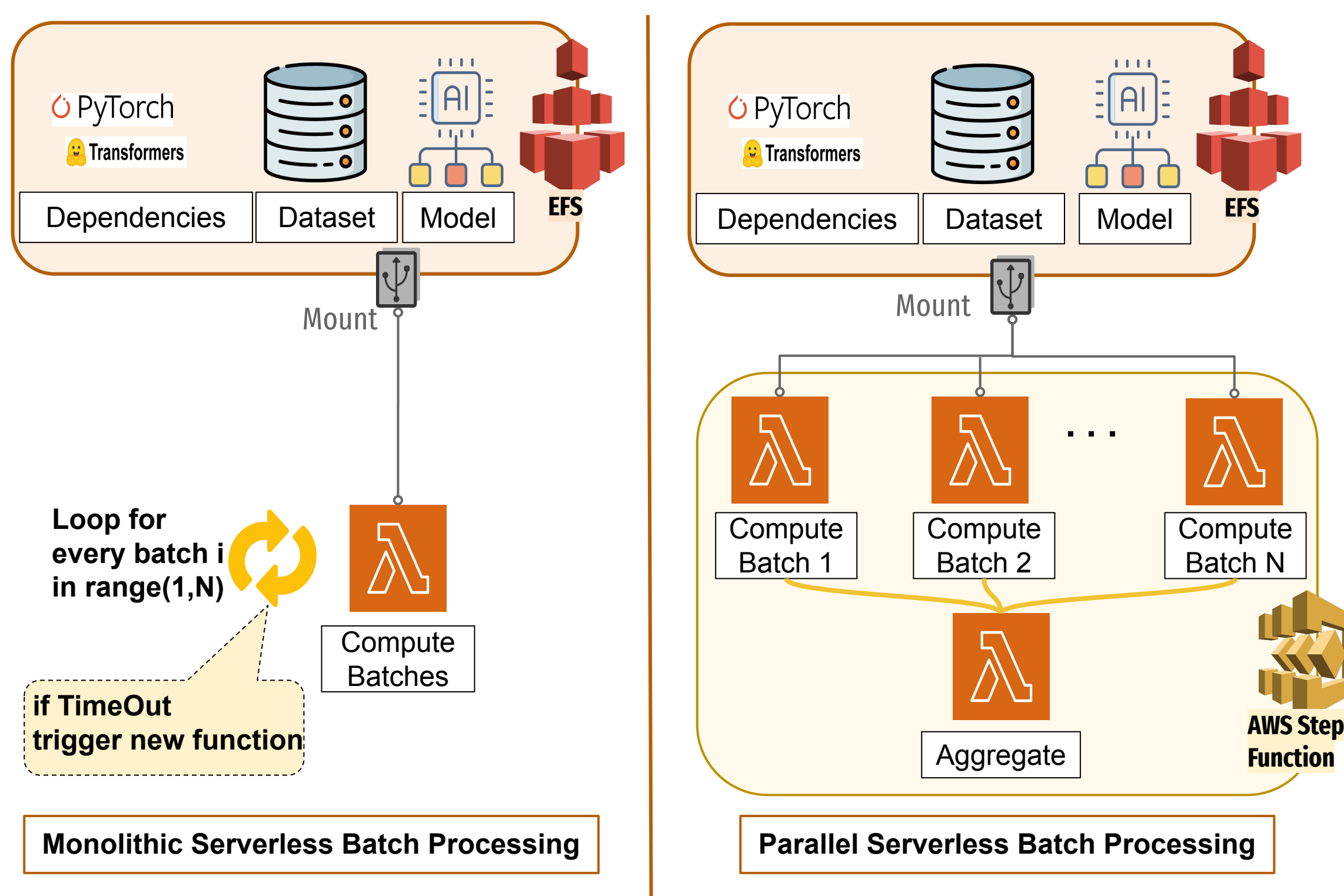
 **Serverless** is a cloud computing model where developers write and deploy code without managing infrastructure, as the cloud provider handles provisioning, scaling, and resource management automatically.

## Contribution

We prove that **ML inference task parallelization in serverless environments** can:

- Deliver **95% faster** results than monolithic.
- Maintain **the same cost as the monolithic** approach.

## Approach



## Experimental Setup

### 1. Dataset & Model:

- **Dataset:** IMDb (25,000 movie reviews).
- **Model:** DistilBERT (66M parameters).

### 2. Processing Approaches:

- **Monolithic Batch Processing:** Sequentially processes all batches in one serverless function.
- **Parallel Batch Processing:**
  - Decomposes dataset into smaller batches.
  - Processes each batch independently using serverless functions.
  - Orchestrates using AWS Step Functions.

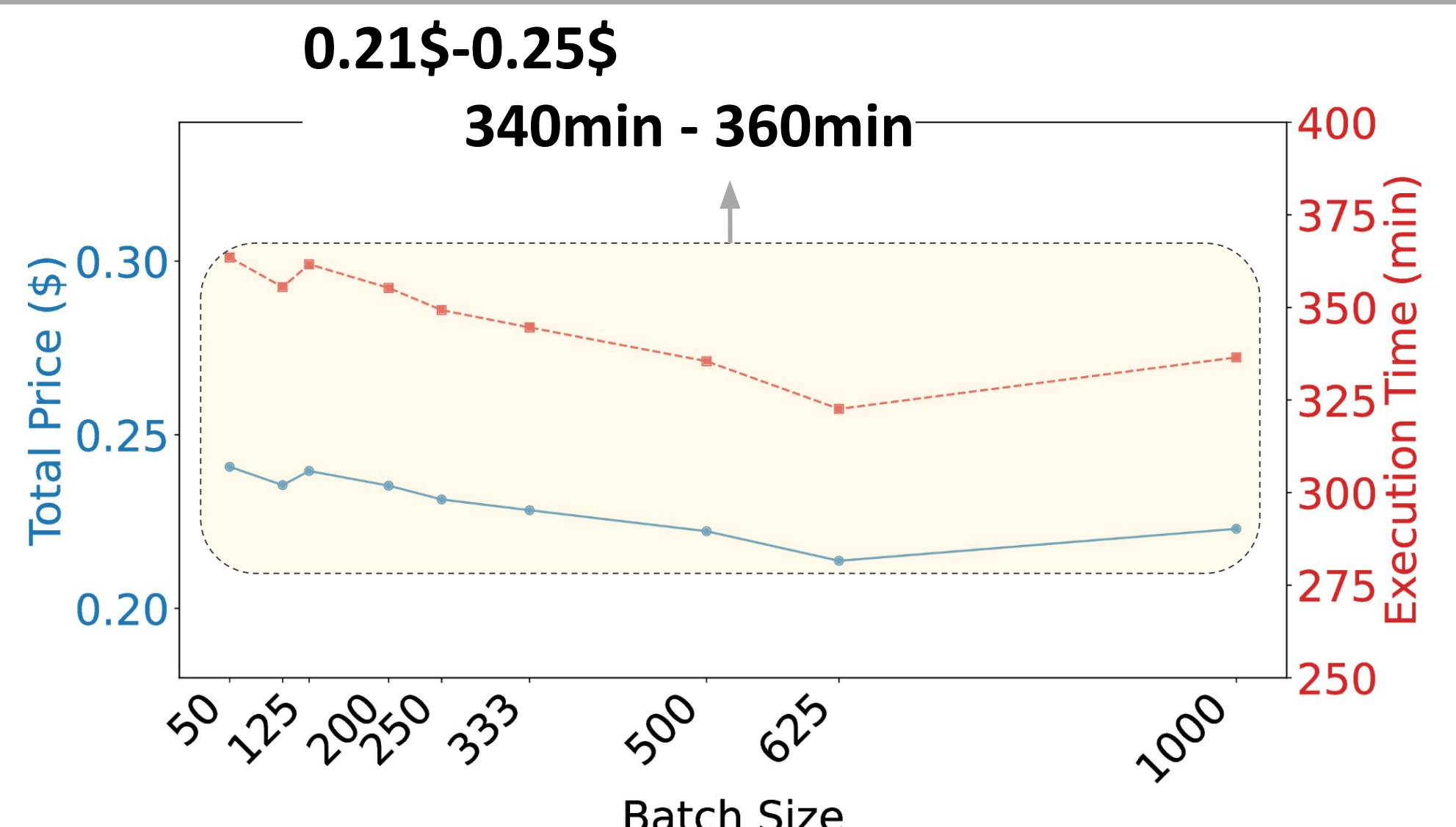
### 3. Evaluation Metrics:

Execution Time & Total Cost & Scalability (Performance with increasing batch sizes)

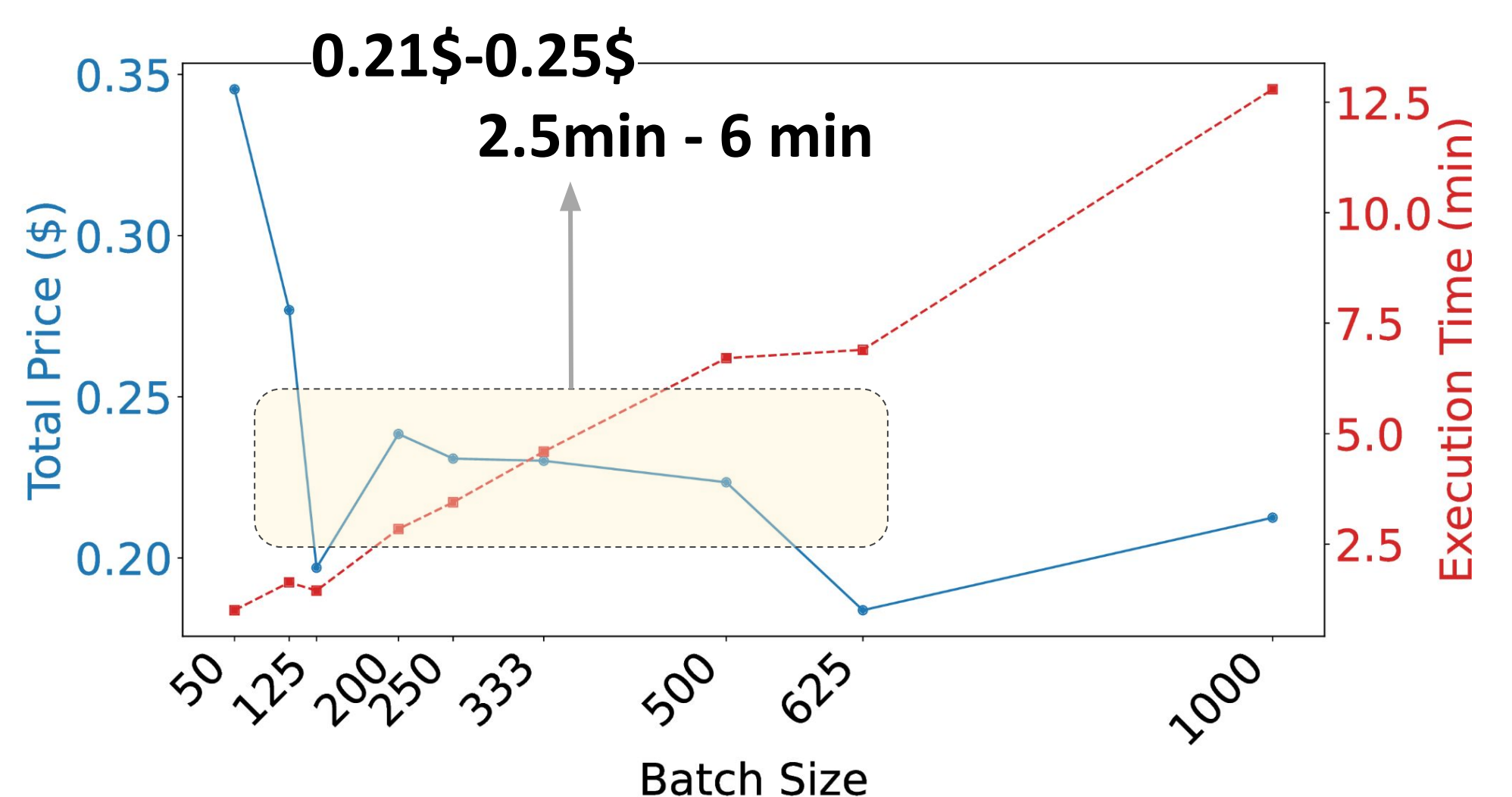
$$\text{Cost}_{\text{function}} = \text{Execution Time}_{\text{ms}} \times \text{Price}_{\text{per ms}}$$

$$\text{Price}_{\text{per ms}} = \text{Function}(\text{RAM}_{\text{allocated}})$$

## Results



(A) Monolithic serverless ML inference



(B) Parallel serverless ML inference

## Key Insights

- Serverless functions are ideal for **short, stateless operations**; breaking monolithic tasks into parallel functions aligns with serverless design principles.
- Inference operations require loading the model's weights into memory, which significantly outweighs the size of the input data. As a result, **memory usage remained consistent in both approaches, leading to stable costs**