CANOPIE-HPC

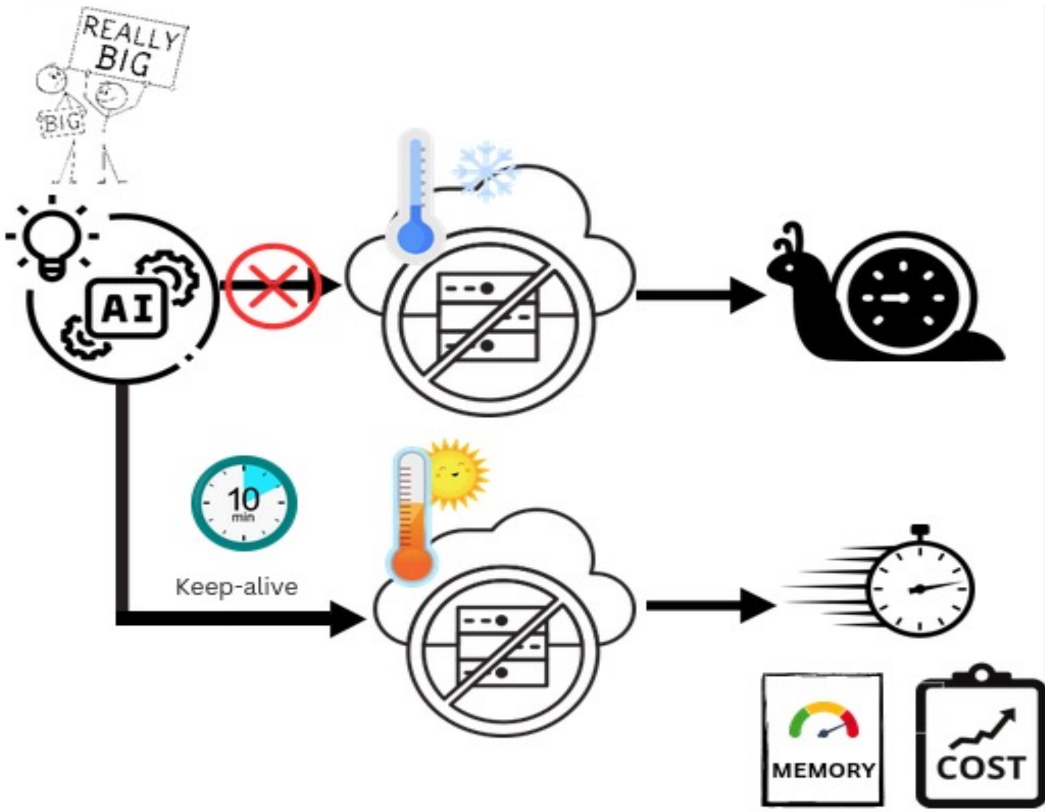
PULSE: Using Mixed-Quality Models for Reducing Serverless Keep-alive Cost

Kausalya Sankaranarayanan,
Rohan Basu Roy, Devesh Tiwari

Northeastern University

Motivation

Machine learning models consume large memory (300-3500MB).



To avoid cold starts,serverless providers keep functions alive for a fixed duration(typically 10 minutes).

The 10-minute fixed keep-alive policy incurs significant memory and cost without adapting to invocation likelihood.

Opportunity I. Machine learning models may come in various variants.

High-Quality Models: Deliver high accuracy but result in longer service times and increased keep-alive costs.

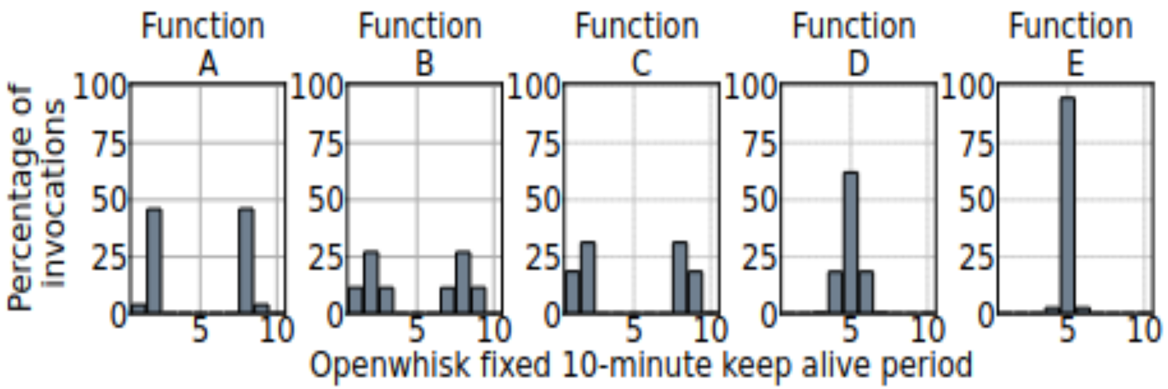
Model	Service Time (with Warmup) (sec)	Keep Alive Cost (cents/hour)	Accuracy (Percent)
GPT-Small	12.90	11.7	87.65
GPT-Medium	22.50	22.57	92.35
GPT-Large	23.66	41.71	93.45
BERT-Small	1.09	4.392	79.6
BERT-Large	2.21	6.12	82.1
DenseNet-121	1.09	3.46	74.98
DenseNet-169	1.38	3.53	76.2
DenseNet-201	1.65	4.07	77.42

Low-Quality Models: Maintain lower cost and faster response times, but at the expense of reduced accuracy.

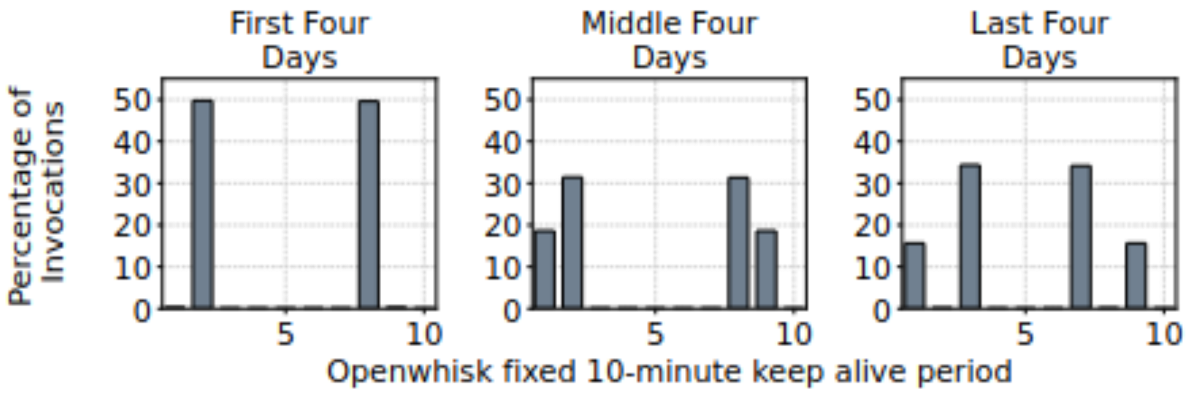
We can combine these model variants during the keep-alive period to reduce both keep-alive cost and memory consumption.

Challenges in Combining Machine Learning Models of Varying Quality on Serverless Execution Platforms

Challenge I. Serverless functions exhibit dynamic invocation patterns.



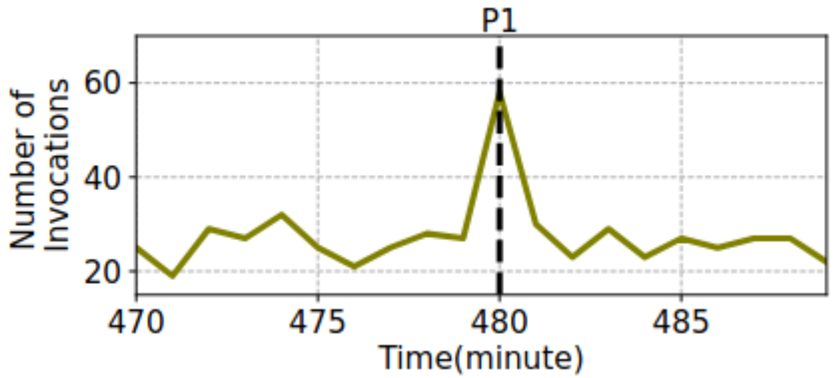
Diverse inter-arrival patterns are observed among various functions.

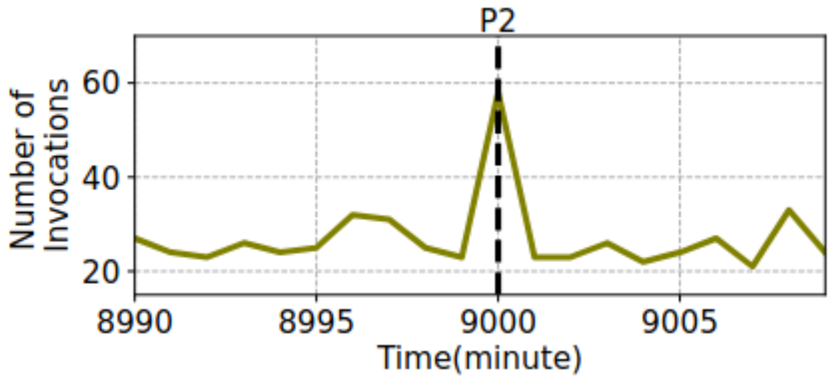


Different inter-arrival time patterns are observed across different periods for the same function.

Data Source: Azure Functions Trace

Challenge II. The existence of peak invocation periods.





	Service Time (sec)	Keep-alive Cost (USD)	Accuracy (Percent)
All High Quality	1799.49	0.86	77.81
All Low Quality	902.38	0.39	71.41
Random High Quality Low Quality	1246.05	0.61	76.13
Intelligent Solution	1661.80	0.78	76.85

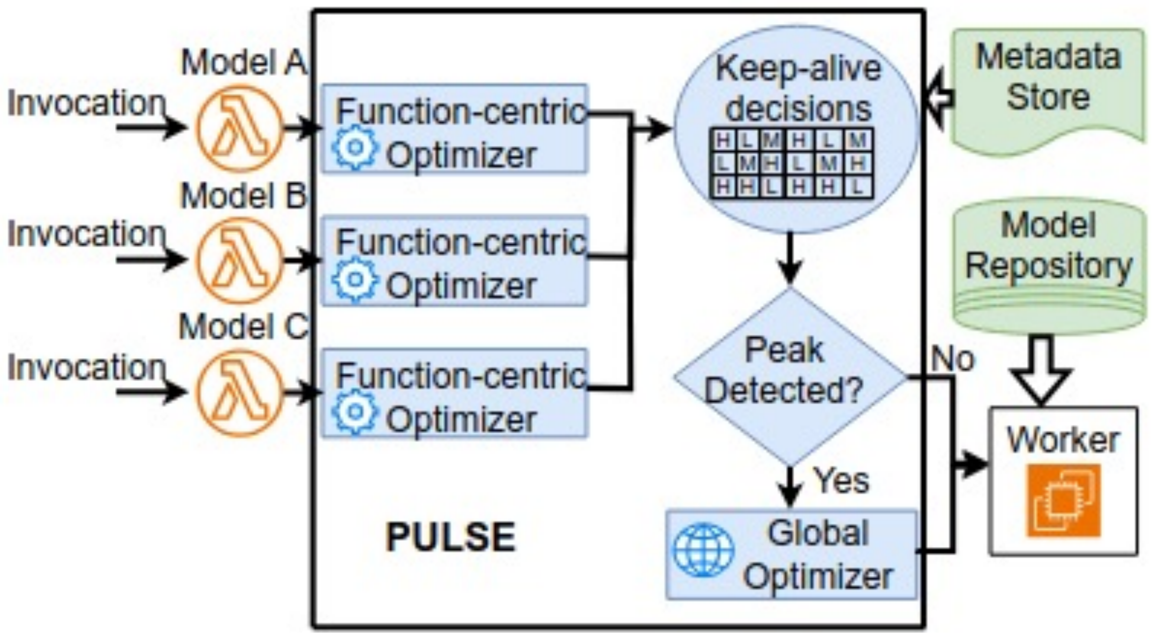
	Service Time (sec)	Keep-alive Cost (USD)	Accuracy (Percent)
All High Quality	1771.12	0.9	78.01
All Low Quality	912.94	0.40	71.62
Random High Quality Low Quality	1246.92	0.62	76.26
Intelligent Solution	1648.79	0.78	77.02

The 10-minute fixed keep-alive policy following a peak incurs substantial costs. This cost can be reduced by mixing models of different qualities, but it must be done strategically, considering both performance and user experience.

PULSE: Key Ideas and Design

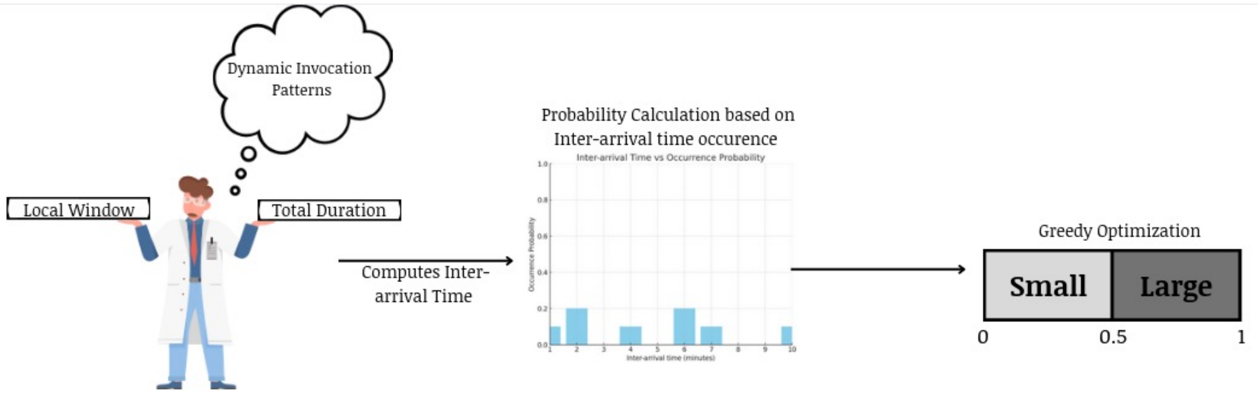
Overview of PULSE

- Individual Function Optimization: Uses historical data and greedy optimization to select model variants to be kept-alive in the 10-minute period following an invocation, reducing keep-alive costs.
- Cross-Function Optimization: During peaks, downgrades functions based on utility values from accuracy, downgrade history, and invocation probabilities.



Probability-based optimization can be used to handle changing invocation patterns.

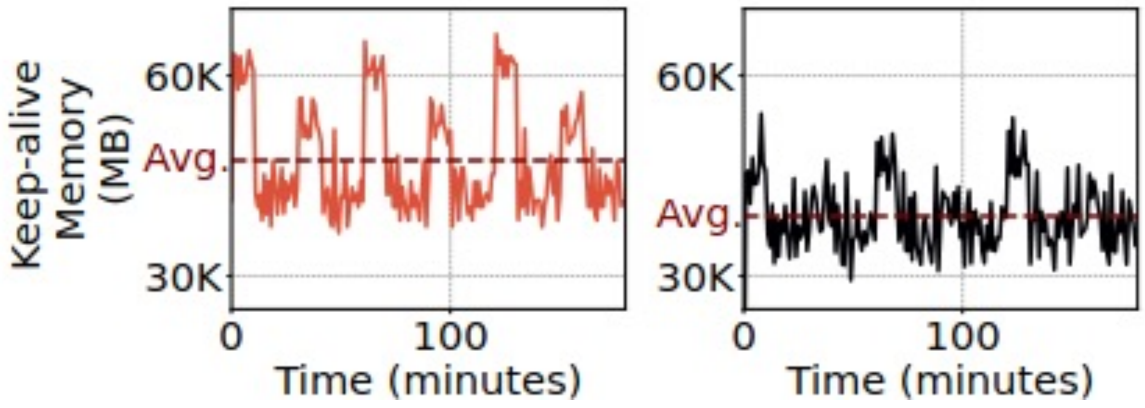
Minimizes overhead, making it ideal for serverless environments handling millions of invocations per time period.



The greedy optimization can be tuned based on providers available resources and needs .

The general principle of keeping the higher-accuracy variants alive for higher probabilities should be followed.

Reduces Keep-Alive Memory, But Peaks Persist



Invocation and keep-alive memory are shared, so keep-alive decisions must account for this.

During resource contention, models can be downgraded.

Unbiased function downgrades are needed to flatten keep-alive memory peaks.

The AWS pricing mentioned in the paper has a typo. The correct rate used is $0.00001667 per GB-second.

Cross-function Optimization
Unbiased downgrading of functions

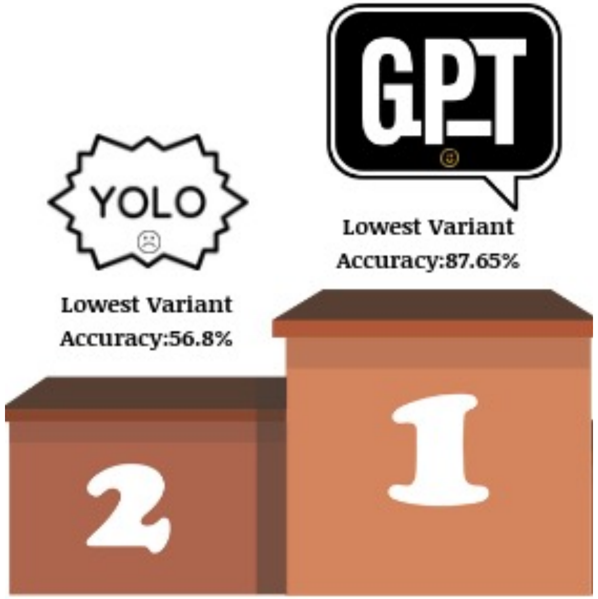
Once a peak is determined we have the following details: probability of invocation and the performance meta data of the model variants.

Using this the downgrade decision process has to be built such that it:

- Evaluates the impact of downgrading on overall performance.
- Doesn't negatively impact user experience .

Accuracy Improvement is Important, but Cannot be the Only Metric to Evaluate Keep-Alive Decisions

- We calculate the accuracy improvement of the chosen keep-alive variant compared to the next lower accuracy variant.
- If the chosen variant is the lowest accuracy option, the accuracy improvement is simply the accuracy of that variant in decimal form.



Using accuracy improvement alone may bias model selection, favoring higher accuracy models like GPT over YOLO; therefore, we introduce a new component called "Priority" to address this.

Priority

A count of how many times each function has be downgraded is maintained in the Priority Structure.

During a peak, this count is normalized to create a priority value for each function.

The function with most downgrades receives the highest priority value.

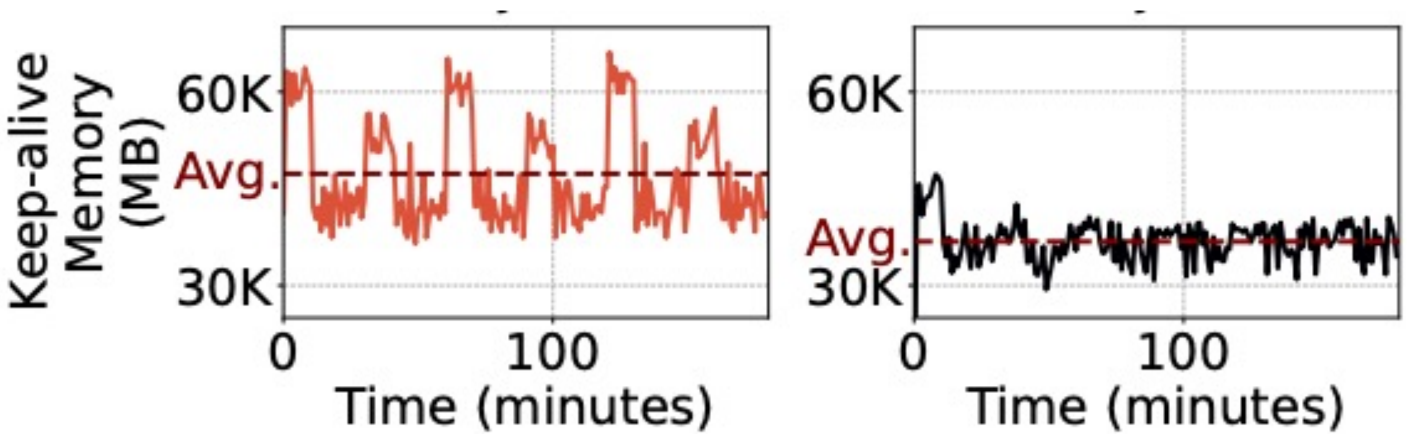
Priority structure

F1:10	F2:1	F3:2	F4:5	F5:4	F6:7

Utility Value Computation

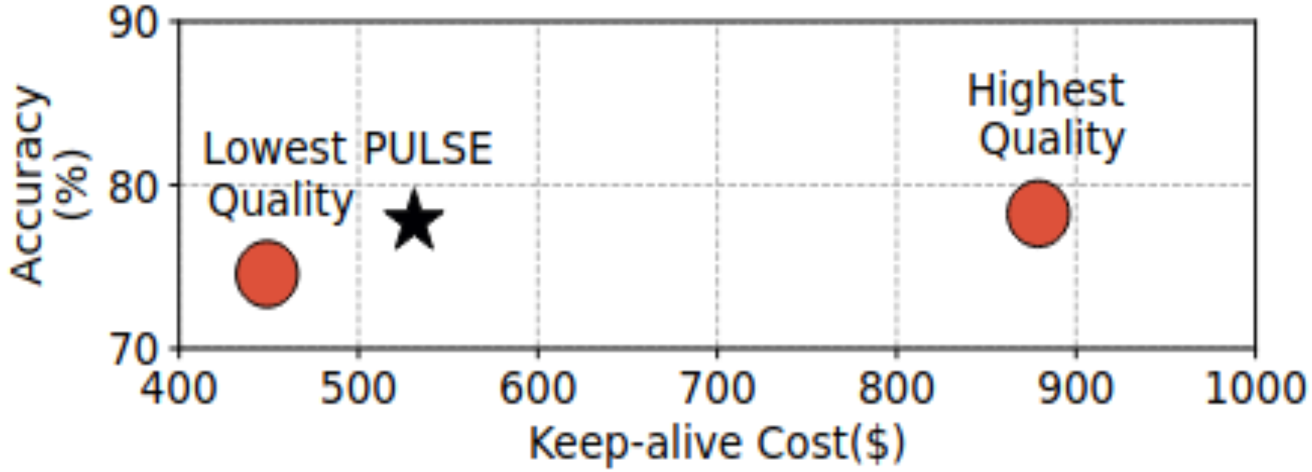
Utility = Accuracy Improvement + Probability of Invocation +Normalized Priority

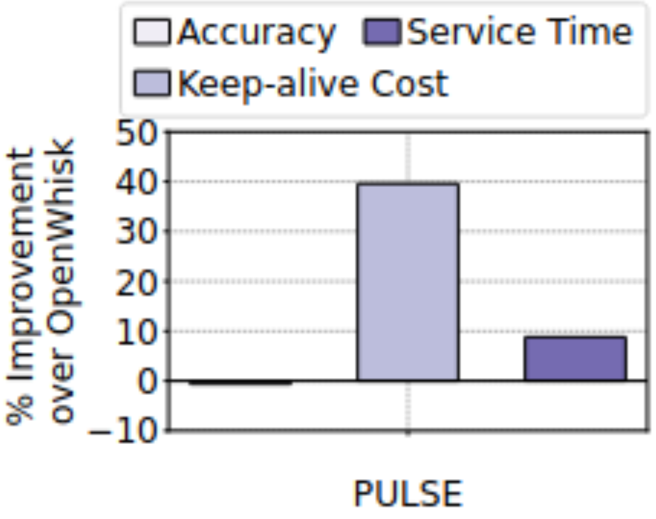
- All the components range between 0 and 1.
- The function with lowest utility is downgraded.
- This is an iterative process that repeats till the peak is flattened.



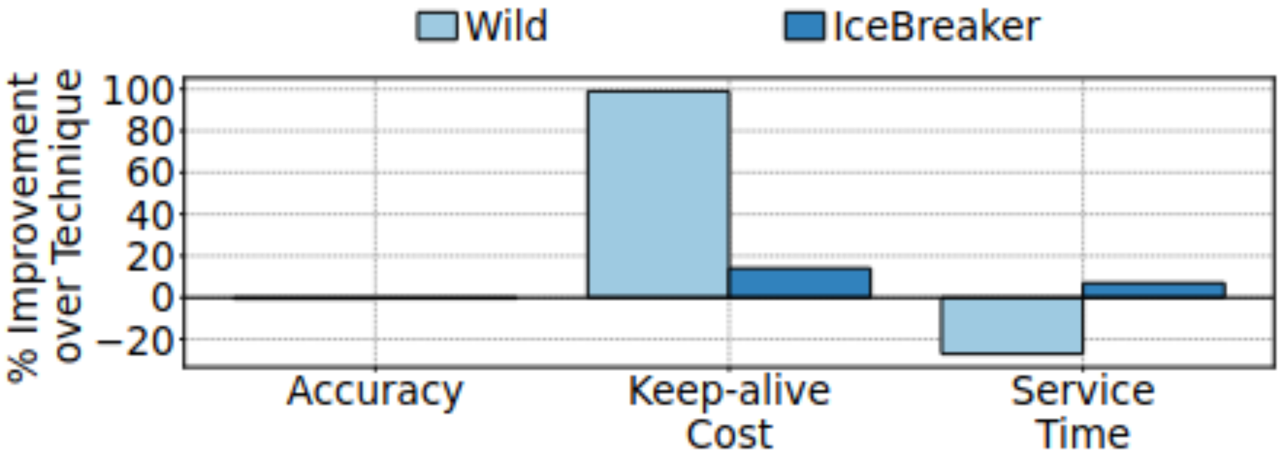
PULSE: Key Results

PULSE shows better performance over OpenWhisk 10-minute fixed keep alive policy

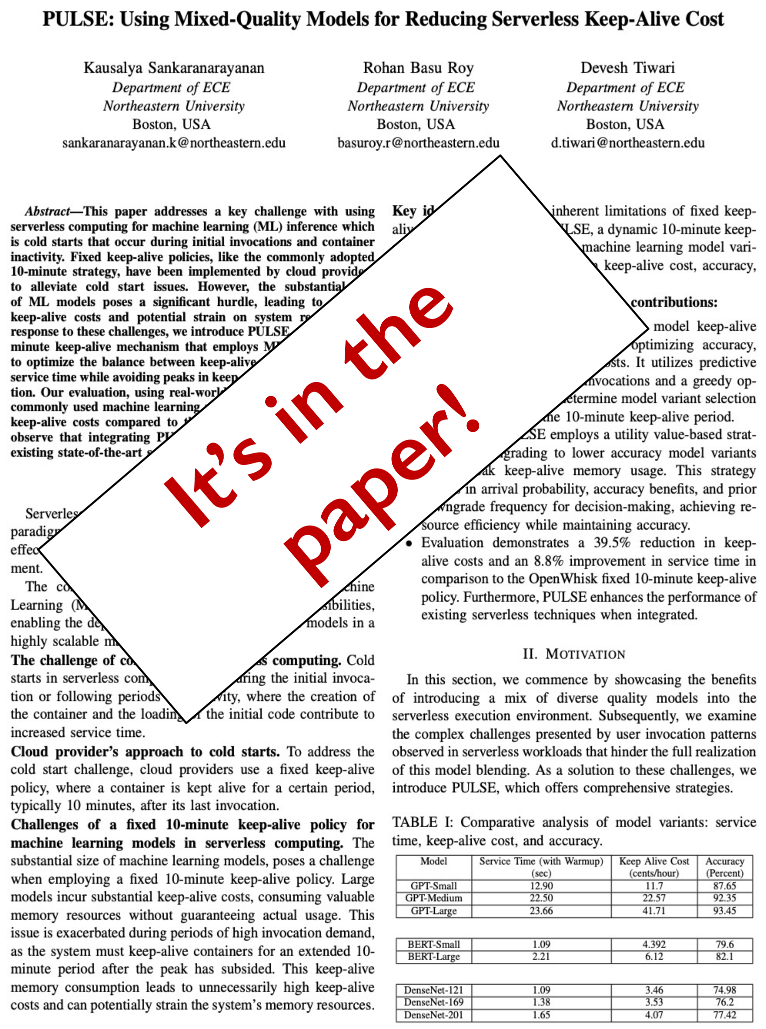




- PULSE achieves accuracy comparable to OpenWhisk's 10-minute fixed keep-alive policy, but at a lower cost.
- PULSE can be integrated with existing state-of-the-art techniques.



PULSE Summary of Contributions



PULSE: Using Mixed-Quality Models for Reducing Serverless Keep-Alive Cost

Kausalya Sankaranarayanan
Department of ECE
Northeastern University
Boston, USA
sankaranarayanan.k@northeastern.edu

Rohan Basu Roy
Department of ECE
Northeastern University
Boston, USA
basuroy.r@northeastern.edu

Devesh Tiwari
Department of ECE
Northeastern University
Boston, USA
d.tiwari@northeastern.edu

Abstract—This paper addresses a key challenge with using
serverless computing for machine learning (ML) inference which
is cold starts that occur during initial invocations and container
inactivity. Fixed keep-alive policies, like the commonly adopted
10-minute strategy, have been implemented by cloud provide
to alleviate cold start issues. However, the substantial
of ML models poses a significant hurdle, leading to
keep-alive costs and potential strain on system r
response to these challenges, we introduce PULSE
minute keep-alive mechanism that employs M
to optimize the balance between keep-alive
service time while avoiding peaks in keep
tion. Our evaluation, using real-worl
commonly used machine learning
keep-alive costs compared to
observe that integrating PI
existing state-of-the-art

Serverles
paradig
effec
ment.

The co
Learning (M
enabling the de
highly scalable m
The challenge of co **s computing.** Cold
starts in serverless com ring the initial invoca-
tion or following periods ivity, where the creation of
the container and the loadin the initial code contribute to
increased service time.

Cloud provider's approach to cold starts. To address the cold start challenge, cloud providers use a fixed keep-alive policy, where a container is kept alive for a certain period, typically 10 minutes, after its last invocation.

Challenges of a fixed 10-minute keep-alive policy for machine learning models in serverless computing. The substantial size of machine learning models, poses a challenge when employing a fixed 10-minute keep-alive policy. Large models incur substantial keep-alive costs, consuming valuable memory resources without guaranteeing actual usage. This issue is exacerbated during periods of high invocation demand, as the system must keep-alive containers for an extended 10-minute period after the peak has subsided. This keep-alive memory consumption leads to unnecessarily high keep-alive costs and can potentially strain the system's memory resources.

Key id inherent limitations of fixed keep-
aliv LSE, a dynamic 10-minute keep-
machine learning model vari-
keep-alive cost, accuracy,

contributions:

model keep-alive
optimizing accuracy,
sts. It utilizes predictive
nvocations and a greedy op-
etermine model variant selection
he 10-minute keep-alive period.
SE employs a utility value-based strat-
grading to lower accuracy model variants
ak keep-alive memory usage. This strategy
in arrival probability, accuracy benefits, and prior
wngrade frequency for decision-making, achieving re-
source efficiency while maintaining accuracy.

- Evaluation demonstrates a 39.5% reduction in keep-alive costs and an 8.8% improvement in service time in comparison to the OpenWhisk fixed 10-minute keep-alive policy. Furthermore, PULSE enhances the performance of existing serverless techniques when integrated.

II. MOTIVATION

In this section, we commence by showcasing the benefits of introducing a mix of diverse quality models into the serverless execution environment. Subsequently, we examine the complex challenges presented by user invocation patterns observed in serverless workloads that hinder the full realization of this model blending. As a solution to these challenges, we introduce PULSE, which offers comprehensive strategies.

TABLE I: Comparative analysis of model variants: service time, keep-alive cost, and accuracy.

Model	Service Time (with Warmup) (sec)	Keep Alive Cost (cents/hour)	Accuracy (Percent)
GPT-Small	12.90	11.7	87.65
GPT-Medium	22.50	22.57	92.35
GPT-Large	23.66	41.71	93.45
BERT-Small	1.09	4.392	79.6
BERT-Large	2.21	6.12	82.1
DenseNet-121	1.09	3.46	74.98
DenseNet-169	1.38	3.53	76.2
DenseNet-201	1.65	4.07	77.42

✓ Miscellaneous design & implementation considerations.

✓ Overhead and sensitivity analysis of PULSE.

✓ More about PULSE design and optimizations.

PULSE

Reducing keep-alive cost and memory consumption in machine learning serverless inference by using different variants.

Contact
Kausalya Sankaranarayanan
sankaranarayanan.k@northeastern.edu

We thank the anonymous reviewers for their constructive feedback. This work was supported by Northeastern University, NSF Award 191601 and 2124897