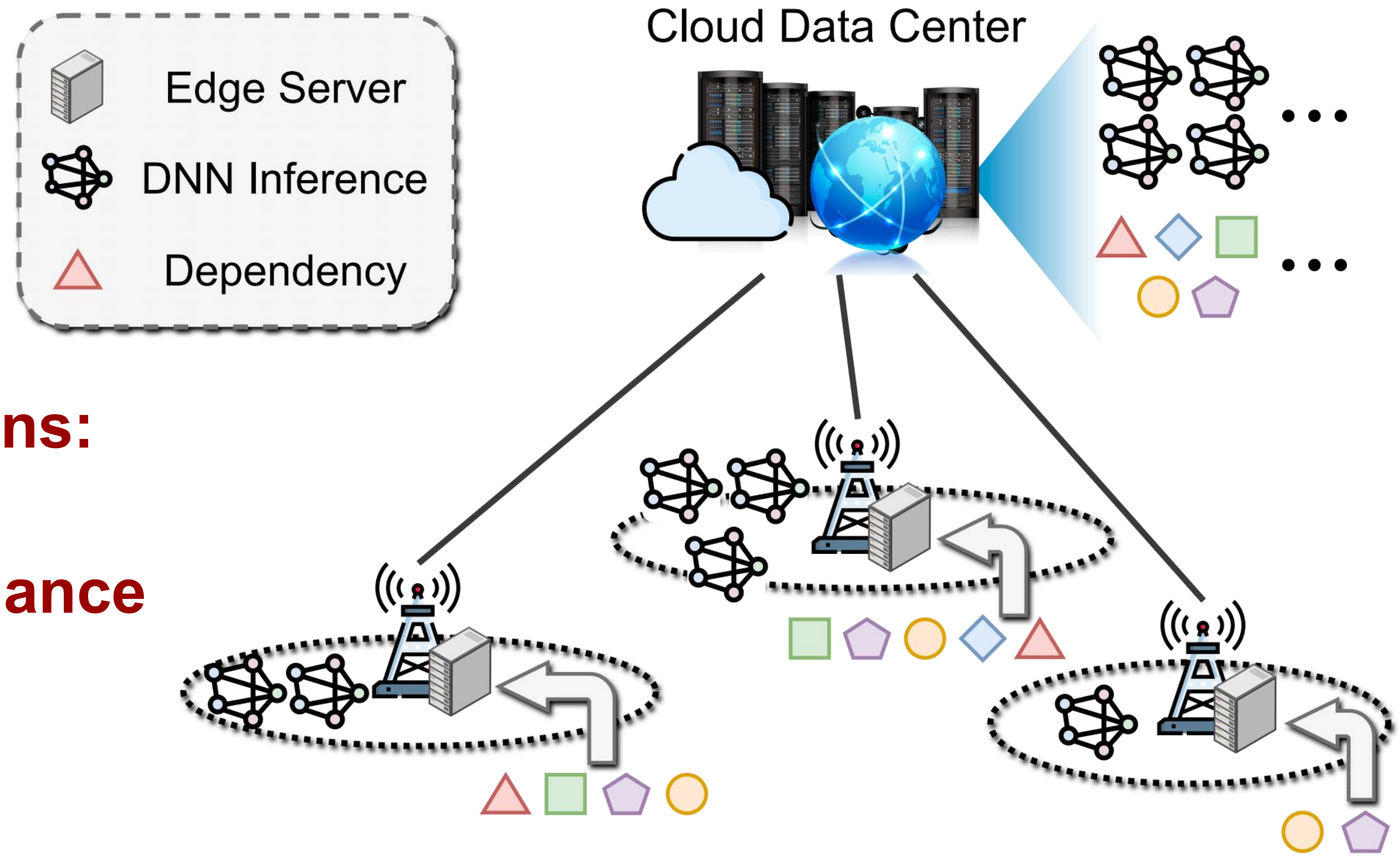
An Inference Acceleration Approach for Boosting DNN Cold Start in Cloud-Edge Computing

Hao Tian[1], *Haolong Xiang*[2], *Tingtong Zhu*[1], *Siyuan Wu*[1], *Cheng Chen*[1], *Zheng Li*[1], *Mingxu Jiang*[1], *Wanchun Dou*[1]

NANJING UNIVERSITY

NANJING UNIVERSITY OF INFORMATION SCIENCE & TECHNOLOGY

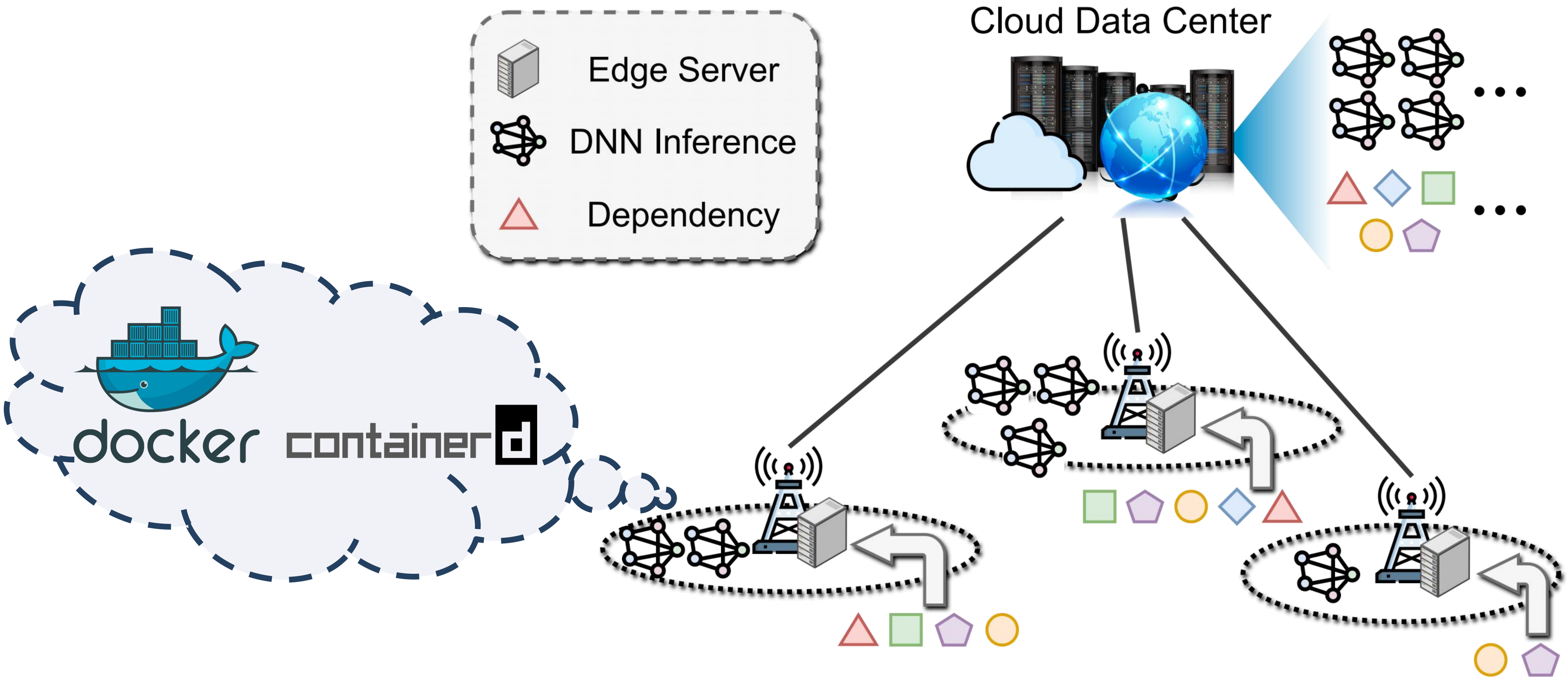
Background



- **Smart Applications:**
  - **IoTs**
  - **Video Surveillance**
  - **Self-driving**
  - **……**

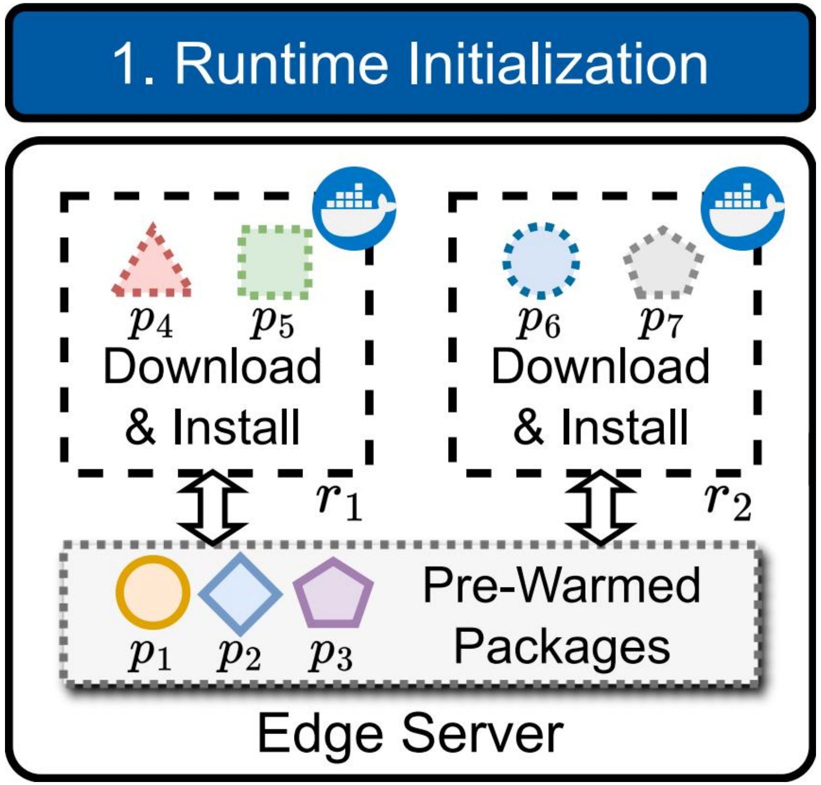
DNN Inference in Cloud-Edge Computing

Background

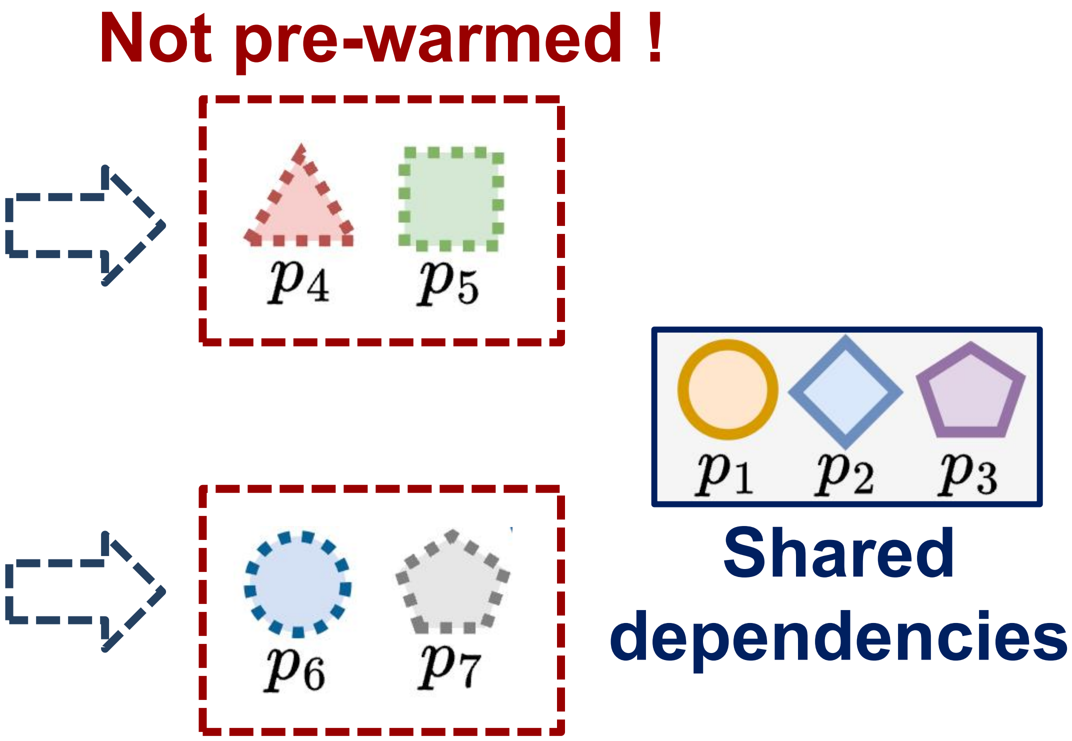


DNN Inference in Cloud-Edge Computing

Motivation

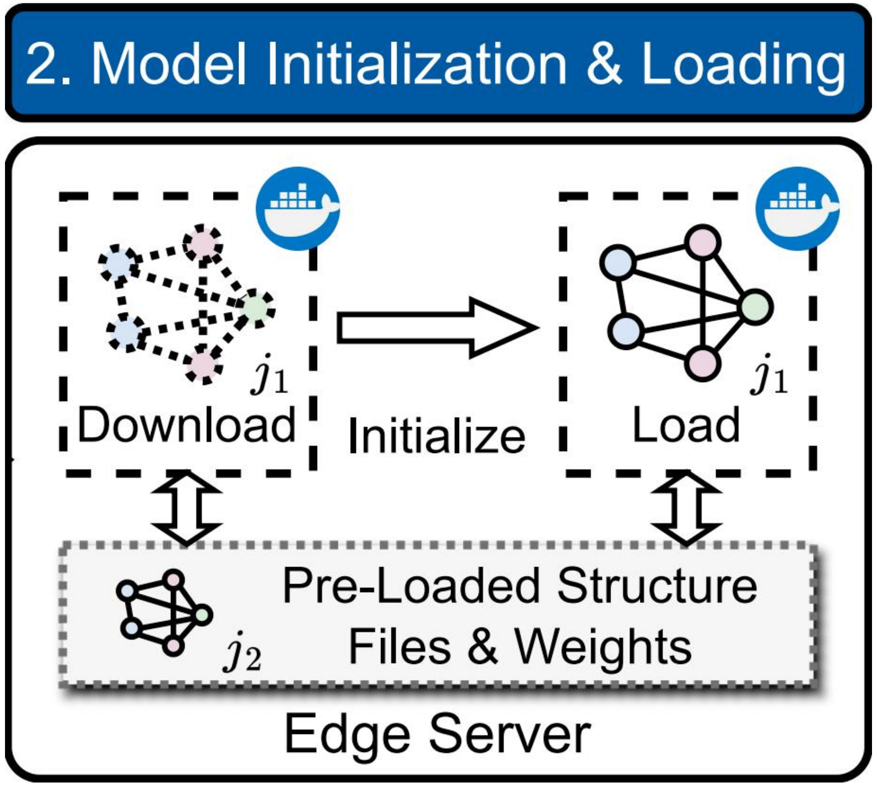


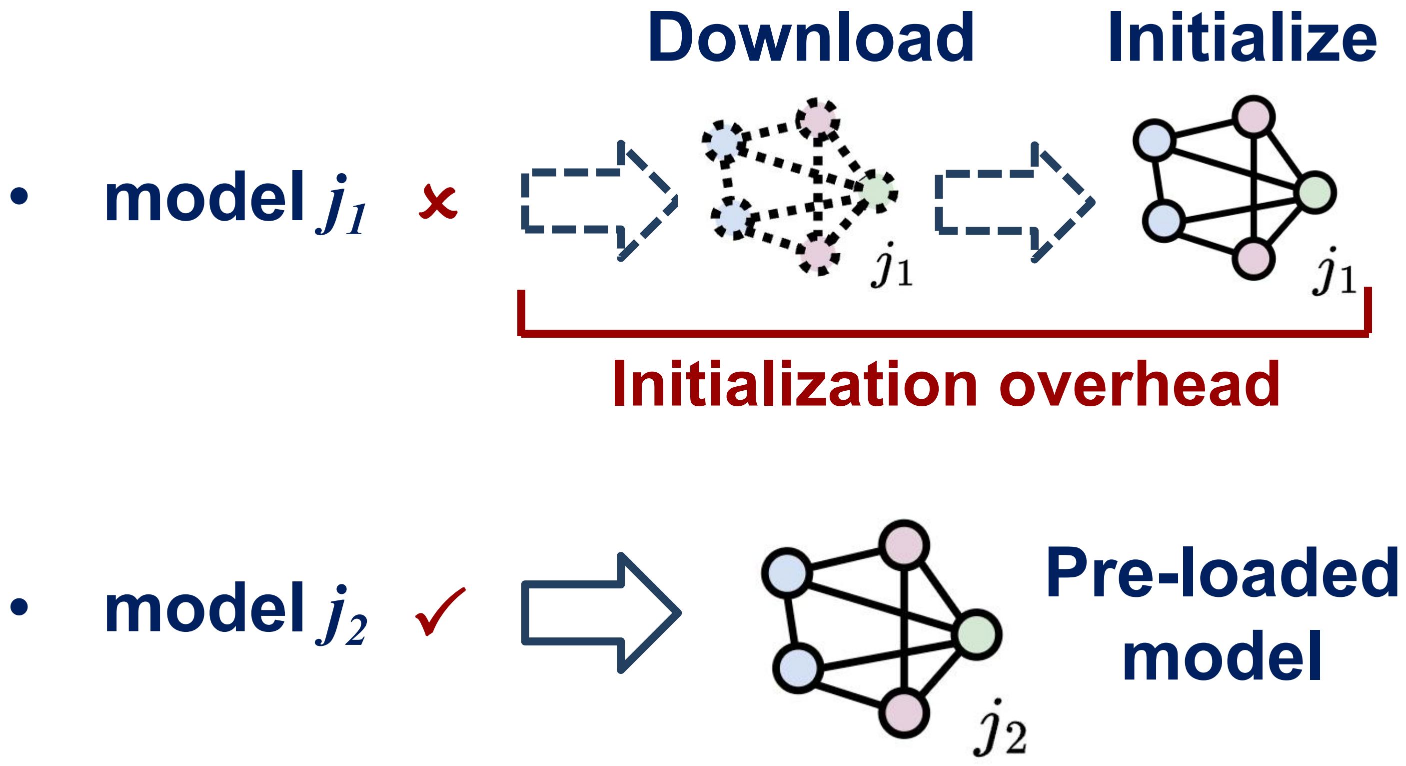
- **request r_1**
- **request r_2**



An Example of Incurred Cold Start for DNN Inference

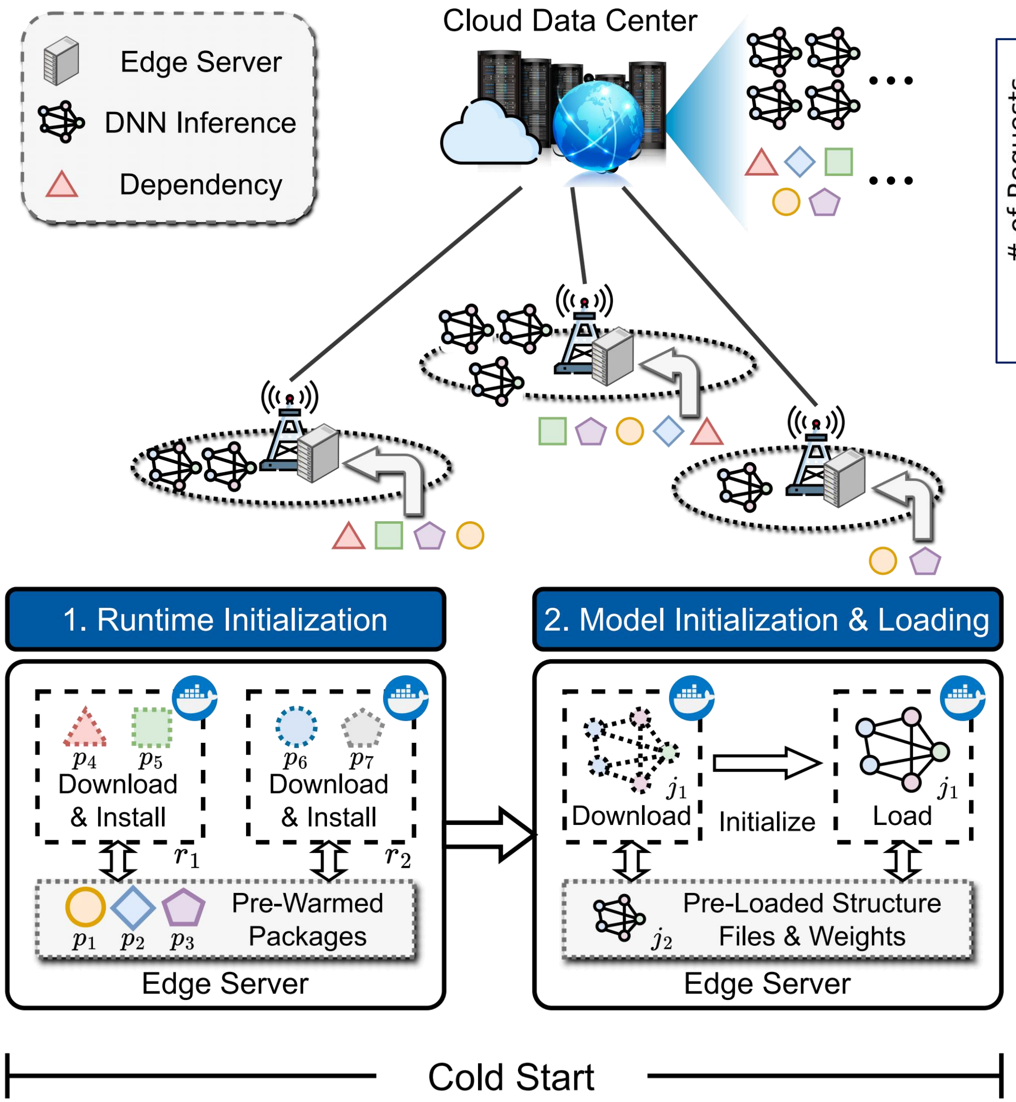
Motivation

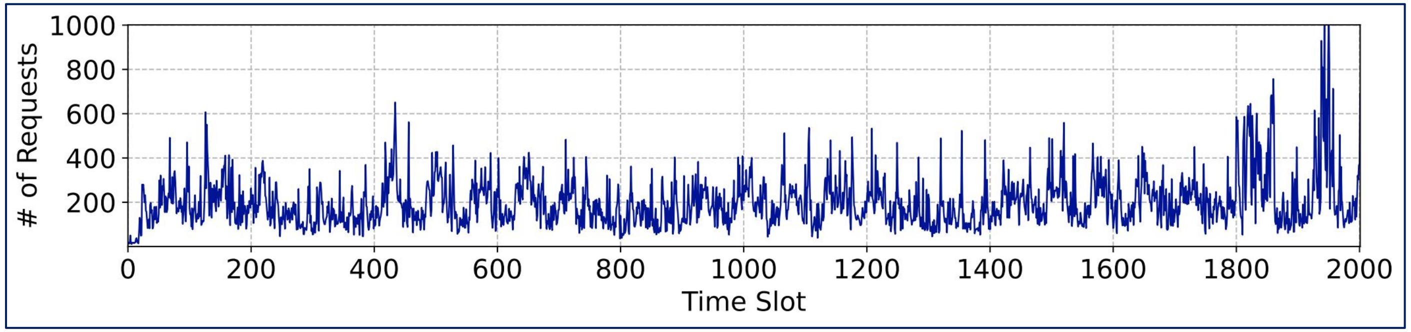




An Example of Incurred Cold Start for DNN Inference

Motivation





- Time-varying request partterns
- Inter-servers resource contention
- Unapropriate initilization settings

How to plan dependencies for edge servers to accelerate DNN inference?

Problem Formulation

$$
\begin{aligned}
\mathcal{P}1: \quad & \min_{\{\alpha,\beta,\zeta,\xi,\mu\}} \boxed{\frac{1}{T}\sum_{t=1}^{T} t^{all}(t)} \\
\textbf{s.t.} \quad & \alpha_n^m \geq 0, \forall n \in \mathcal{N}, \forall m \in \mathcal{M}, \\
& \textstyle\sum_{m \in \mathcal{M}} \alpha_n^m \leq 1, \forall n \in \mathcal{N}, \\
& \beta_n^j \geq 0, \forall n \in \mathcal{N}, \forall j \in \mathcal{J}, \\
& \textstyle\sum_{j \in \mathcal{J}} \beta_n^j \leq 1, \forall n \in \mathcal{N}, \\
& \zeta_p^n, \xi_j^n, \mu_j^n \in \{0,1\}, \forall p \in \mathcal{P}, \forall j \in \mathcal{J}, \forall n \in \mathcal{N}.
\end{aligned}
$$

- The objective is to ***minimize the time-average end-to-end inference time***
- Constrainted by:
  - resource limit
  - dependencies planning

Problem Formulation

$$\mathcal{P}1: \quad \min_{\{\alpha,\beta,\zeta,\xi,\mu\}} \boxed{\frac{1}{T}\sum_{t=1}^{T} t^{all}(t)}$$

$$\begin{aligned}
\textbf{s.t.} \quad & \alpha_n^m \geq 0, \forall n \in \mathcal{N}, \forall m \in \mathcal{M}, \\
& \textstyle\sum_{m\in\mathcal{M}} \alpha_n^m \leq 1, \forall n \in \mathcal{N}, \\
& \beta_n^j \geq 0, \forall n \in \mathcal{N}, \forall j \in \mathcal{J}, \\
& \textstyle\sum_{j\in\mathcal{J}} \beta_n^j \leq 1, \forall n \in \mathcal{N}, \\
& \zeta_p^n, \xi_j^n, \mu_j^n \in \{0,1\}, \forall p \in \mathcal{P}, \forall j \in \mathcal{J}, \forall n \in \mathcal{N}.
\end{aligned}$$

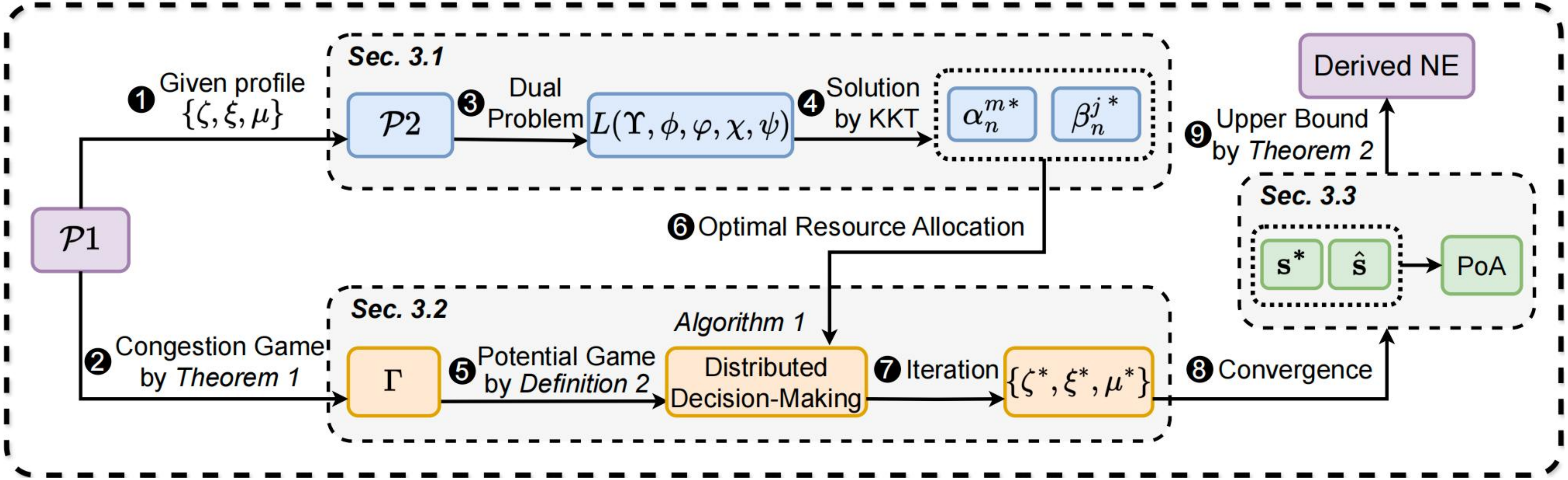
$$\boxed{t^{all}(t) = \sum_{n\in\mathcal{N}} \sum_{j\in\mathcal{J}} \sum_{m\in\mathcal{M}} \underline{t_{n,m}^{up}(t)} + \underline{t_{n,j}^{dp}(t)} + \underline{t_{n,j}^{dm}(t)} + \underline{t_{n,j}^{c}(t)}.}$$

uploading time

startup time

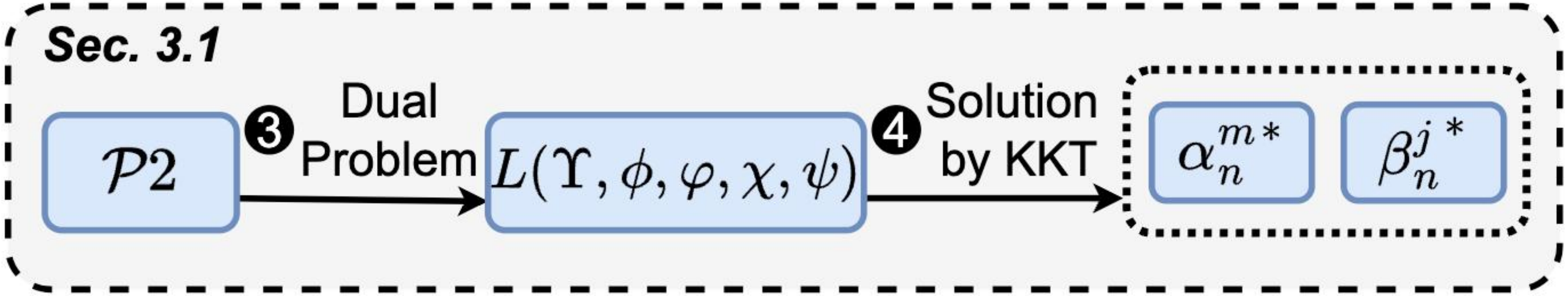
processing time

Algorithm Design



The framework of INAA

Algorithm Design



$$\mathcal{P}1: \min_{\{\alpha,\beta,\zeta,\xi,\mu\}} \frac{1}{T}\sum_{t=1}^{T} t^{all}(t)$$

$$\begin{aligned}
\text{s.t.} \quad & \alpha_n^m \geq 0, \forall n \in \mathcal{N}, \forall m \in \mathcal{M}, \\
& \textstyle\sum_{m\in\mathcal{M}} \alpha_n^m \leq 1, \forall n \in \mathcal{N}, \\
& \beta_n^j \geq 0, \forall n \in \mathcal{N}, \forall j \in \mathcal{J}, \\
& \textstyle\sum_{j\in\mathcal{J}} \beta_n^j \leq 1, \forall n \in \mathcal{N}, \\
& \zeta_p^n, \xi_j^n, \mu_j^n \in \{0,1\}, \forall p \in \mathcal{P}, \forall j \in \mathcal{J}, \forall n \in \mathcal{N}.
\end{aligned}$$

Given Planning Profiles

$$\mathcal{P}2: \min_{\{\alpha,\beta\}} \Psi(\zeta, \xi, \mu, t)$$

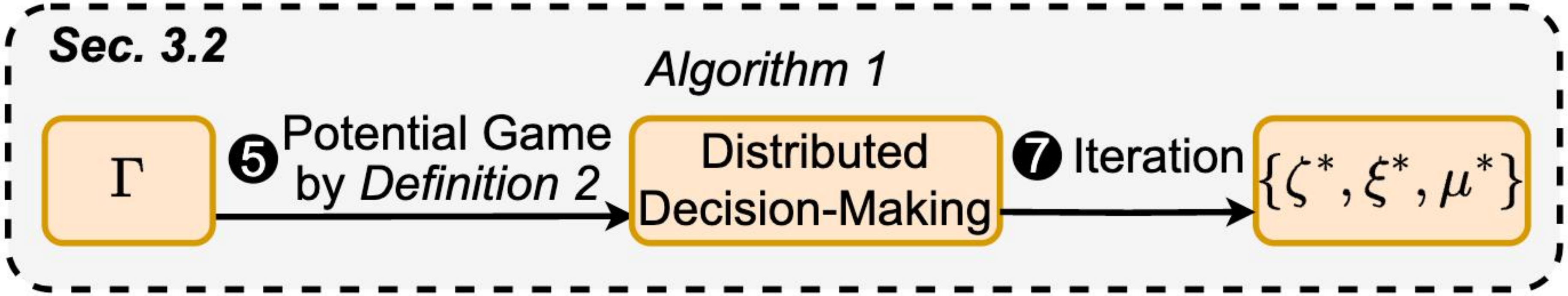
$$\text{s.t.} \quad (2a) - (2d)$$

Solved by KKT Conditions

optimal resource decisions *α and *β***

Design #1: Problem Transformation and Optimal Allocation

Algorithm Design



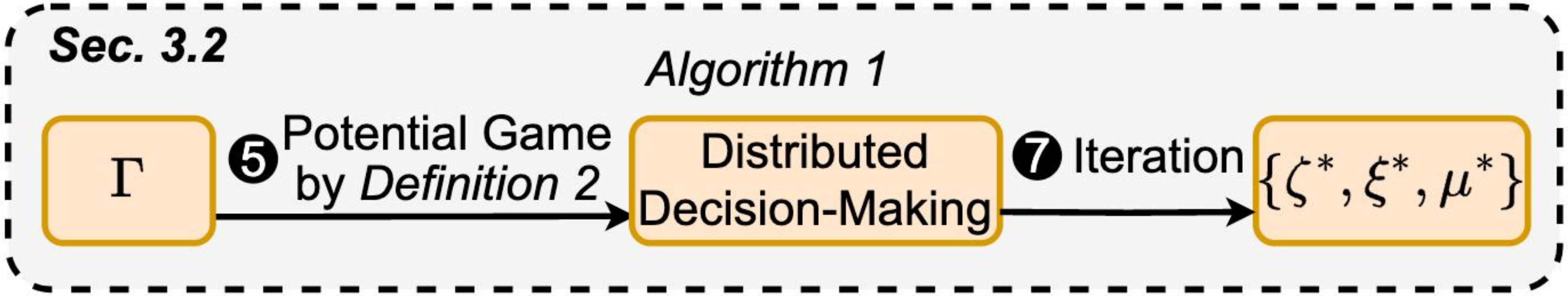
By theorem of congestion game:

$$\Gamma = < M, \{\zeta_n, \xi_n, \mu_n\}_{n \in N}, \{t^{all}\}_{n \in N} >$$

The key goal is to find pure-strategy Nash equilibrium (NE)

***Design #2:* Game Theory-based Inference Acceleration**

Algorithm Design



$$\Gamma = <M, \{\zeta_n, \xi_n, \mu_n\}_{n \in N}, \{t^{all}\}_{n \in N}>$$

Exact Potential Game

$$t^{all}{}_n(s`_n, s_{-n}, t) - t^{all}{}_n(s_n, s_{-n}, t) = \boxed{\Phi(s`_n, s_{-n}, t)} - \Phi(s_n, s_{-n}, t)$$

global potential function

Design #2: Game Theory-based Inference Acceleration

Algorithm Design

Algorithm 1: Distributed resource allocation and inference acceleration in time slot $t \in \mathcal{T}$

```
Initialize randomized decisions of dependencies planning in edge servers;
repeat
    for n ∈ 𝒩 in parallel do
        Calculate the sum of total time under decision (s_n(t), s_{-n}(t)) with
        optimal resource allocation α_n* and β_n*;
        Compute minimal total time with s'_n(t) and report it for contend;
    end
    for n ∈ 𝒩 in parallel do
        if n wins the competition with highest improvement then
            Update the decision with s*_n(t) and s(t) ← {s*_n(t), s'_{-n}(t)};
        end
    end
until no edge server updates its decision;
Return s*(t).
```

Find optimal decision iteratively

Algorithm Design

Algorithm 1: Distributed resource allocation and inference acceleration in time slot $t \in \mathcal{T}$

```
Initialize randomized decisions of dependencies planning in edge servers;
repeat
    for n ∈ N in parallel do
        Calculate the sum of total time under decision (s_n(t), s_{-n}(t)) with
          optimal resource allocation α_n* and β_n*;
        Compute minimal total time with s'_n(t) and report it for contend;
    end
    for n ∈ N in parallel do
        if n wins the competition with highest improvement then
            Update the decision with s*_n(t) and s(t) ← {s*_n(t), s'_{-n}(t)};
        end
    end
until no edge server updates its decision;
Return s*(t).
```

Competition among edge servers

Algorithm Design

Algorithm 1: Distributed resource allocation and inference acceleration in time slot $t \in \mathcal{T}$

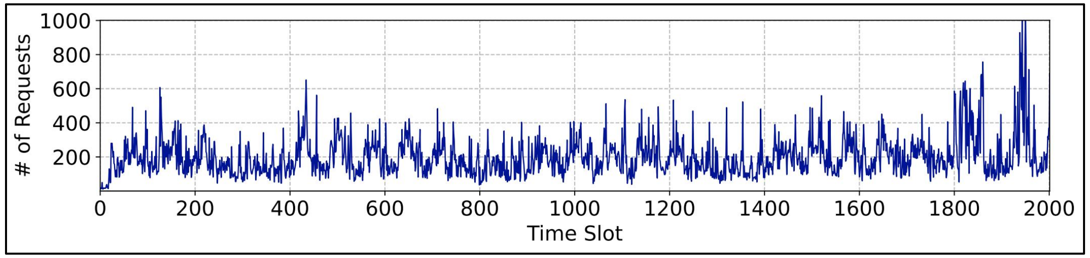
Initialize randomized decisions of dependencies planning in edge servers;
repeat
    for $n \in \mathcal{N}$ *in parallel* **do**
        Calculate the sum of total time under decision $\left(s_n(t), s_{-n}(t)\right)$ with optimal resource allocation $\alpha_n{}^*$ and $\beta_n{}^*$;
        Compute minimal total time with $s'_n(t)$ and report it for contend;
    end
    for $n \in \mathcal{N}$ *in parallel* **do**
        if *n wins the competition with highest improvement* **then**
            Update the decision with $s_n^*(t)$ and $\mathbf{s}(t) \leftarrow \{s_n^*(t), s'_{-n}(t)\}$;
        end
    end
until *no edge server updates its decision*;
Return $\mathbf{s}^*(t)$.

Under finite improvement property (FIP)

Evaluation

- Datasets
  - EUA[1]
  - GPU trace from Alibaba cluster[2]
- DNN Models[3]
  - AlexNet
  - NiN
  - ResNet32
  - VGG16
- Baselines
  - HP[4]
  - LWC[5]

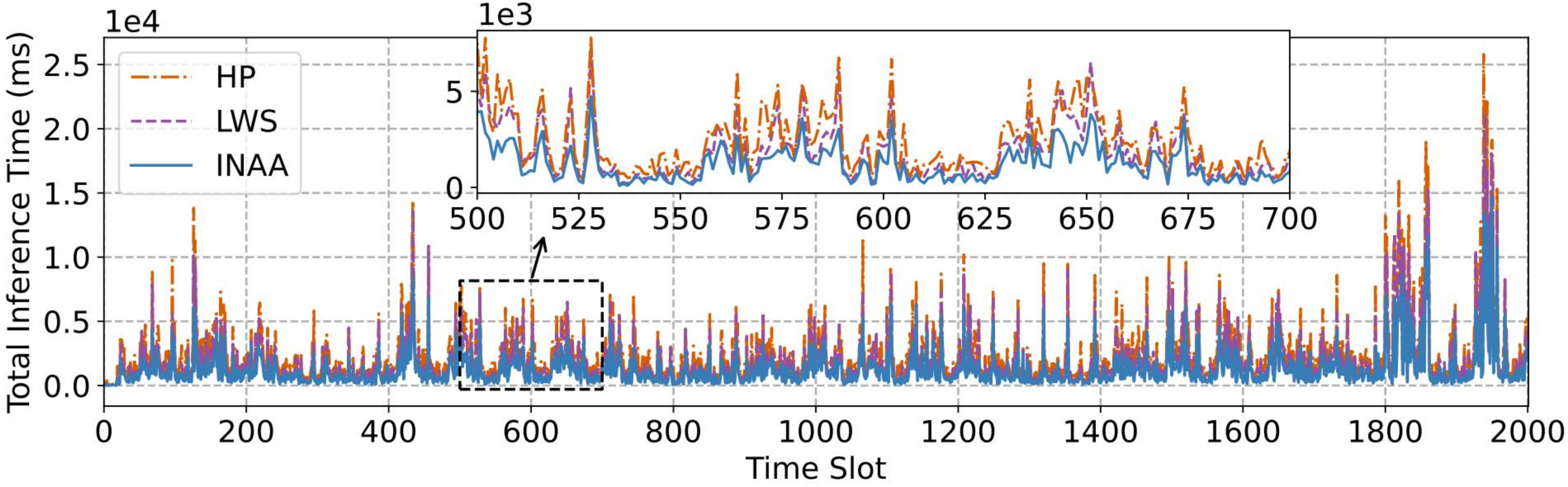
request arrivals



[1] He, Q., et al.: A game-theoretical approach for user allocation in edge computing environment. IEEE Trans. Parallel Distrib. Syst. 31(3), 515–529 (2019)
[2] Weng, Q., et al.: MLaaS in the wild: workload analysis and scheduling in large_x0002_scale heterogeneous GPU clusters. In: 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22) (2022)
[3] Mohammed, T., Joe-Wong, C., Babbar, R., Di Francesco, M.: Distributed inference acceleration with adaptive DNN partitioning and offloading. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, pp. 854–863. IEEE (2020)
[4] Shahrad, M., et al.: Serverless in the wild: characterizing and optimizing the server_x0002_less workload at a large cloud provider. In: 2020 USENIX Annual Technical Con_x0002_ference (USENIX ATC 20), pp. 205–218 (2020)
[5] Sethi, B., Addya, S.K., Ghosh, S.K.: LCS: alleviating total cold start latency in serverless applications with LRU warm container approach. In: Proceedings of the 24th International Conference on Distributed Computing and Networking, pp. 197–206 (2023)

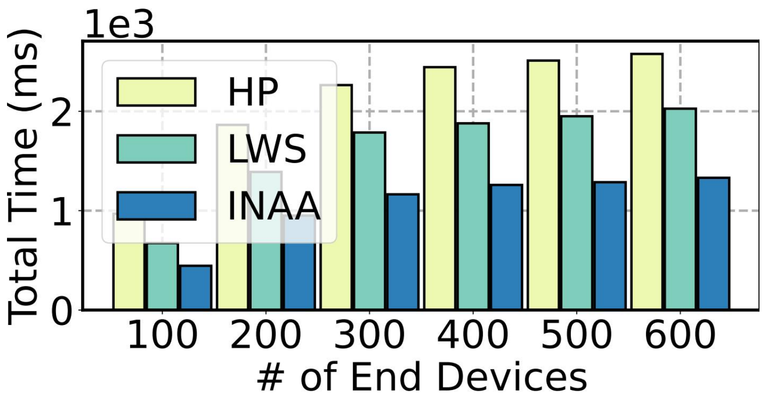
Evaluation

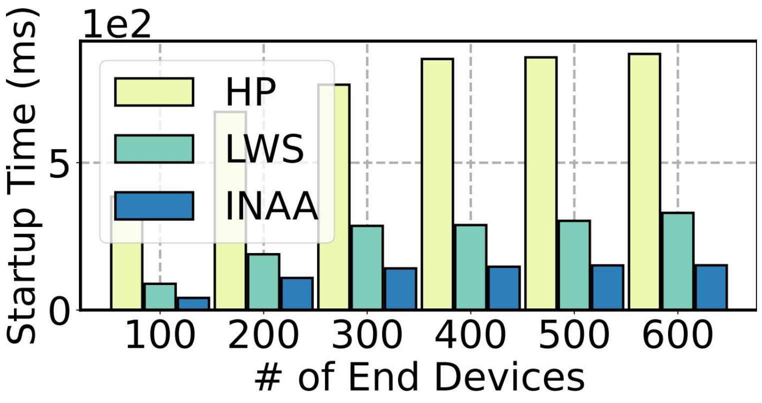
- INAA achieves lower E2E time when the request arrival is at the peak

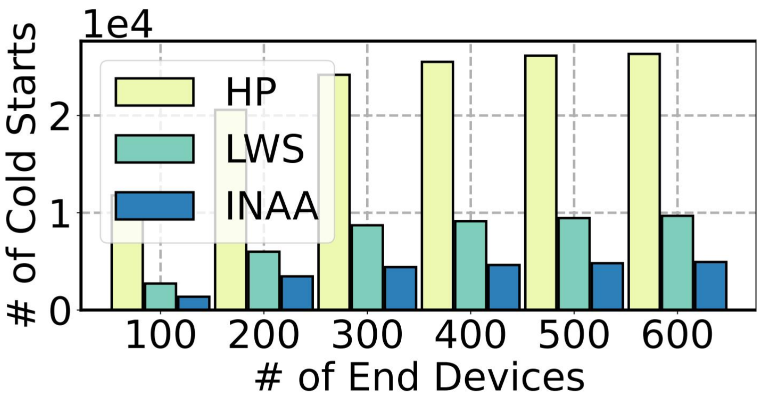


Evaluation

■ **INAA gains the lower startup time by the growing number of devices**

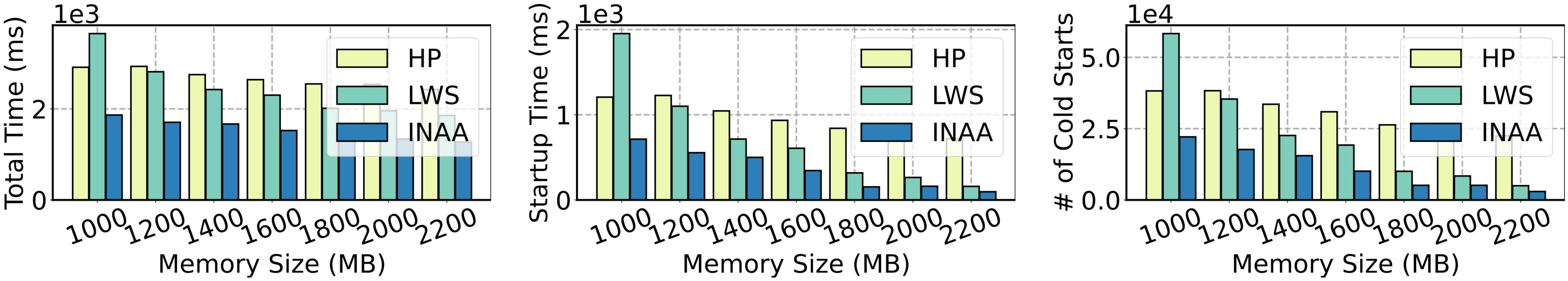






Evaluation

- Up to 58.9% improvement in E2E time under increasing memory size



	Total time (ms)	Startup time (ms)	Cold starts (#)
HP	2414.42	861.96	26969.0
LWC	2056.40	507.71	15744.0
INAA	**1319.09**	**251.14**	**7868.0**

Conclusion

- Our proposed INAA is to jointly improve the startup time and resource utilization
- The original denepdency planning problem is decomposed into sub-problems, solved by convex optimization technique
- INAA achieves minimal execution cost with less cold starts under the distributed decision-making

Thank You!