EcoFaaS: Rethinking the Design of Serverless Environments for Energy Efficiency

ISCA 2024

Jovan Stojkovic, Nikoleta Iliakopoulou, Tianyin Xu, Hubertus Franke*, Josep Torrellas

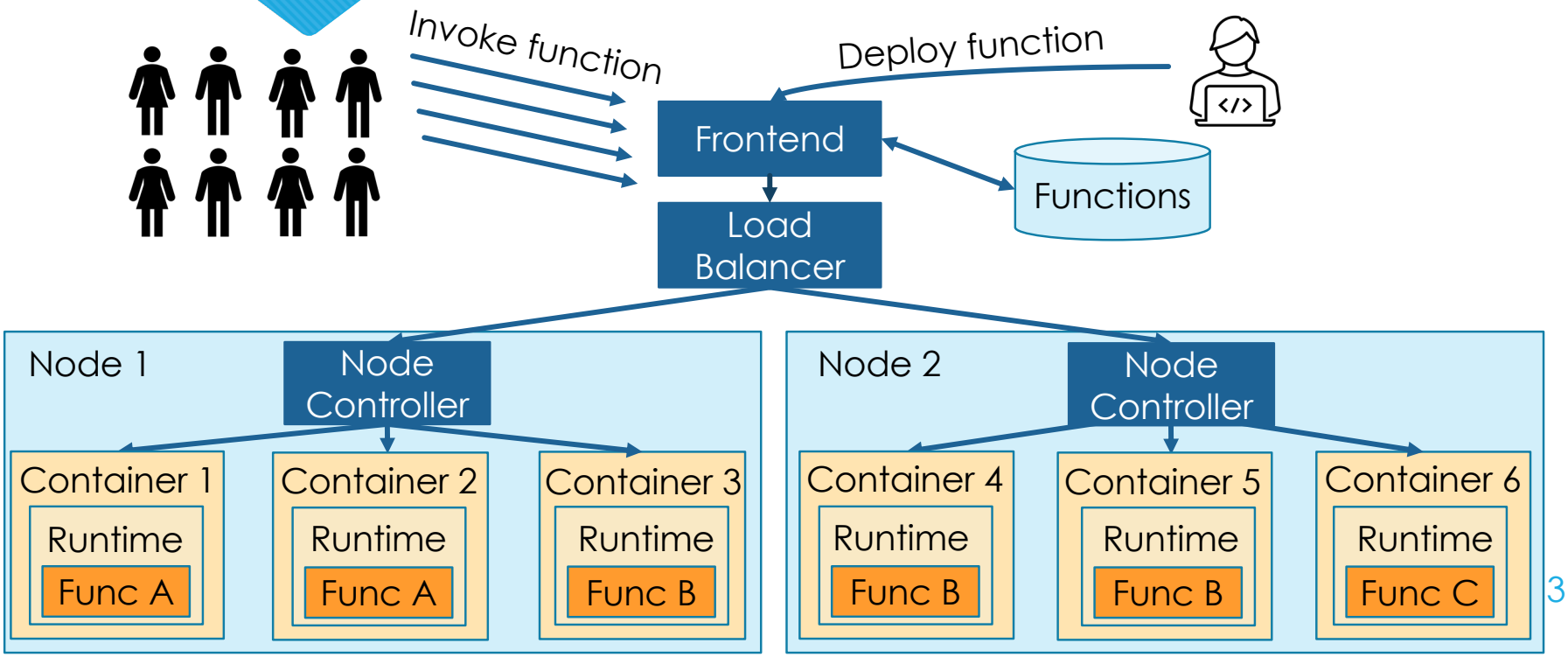
University of Illinois at Urbana-Champaign

*IBM Research

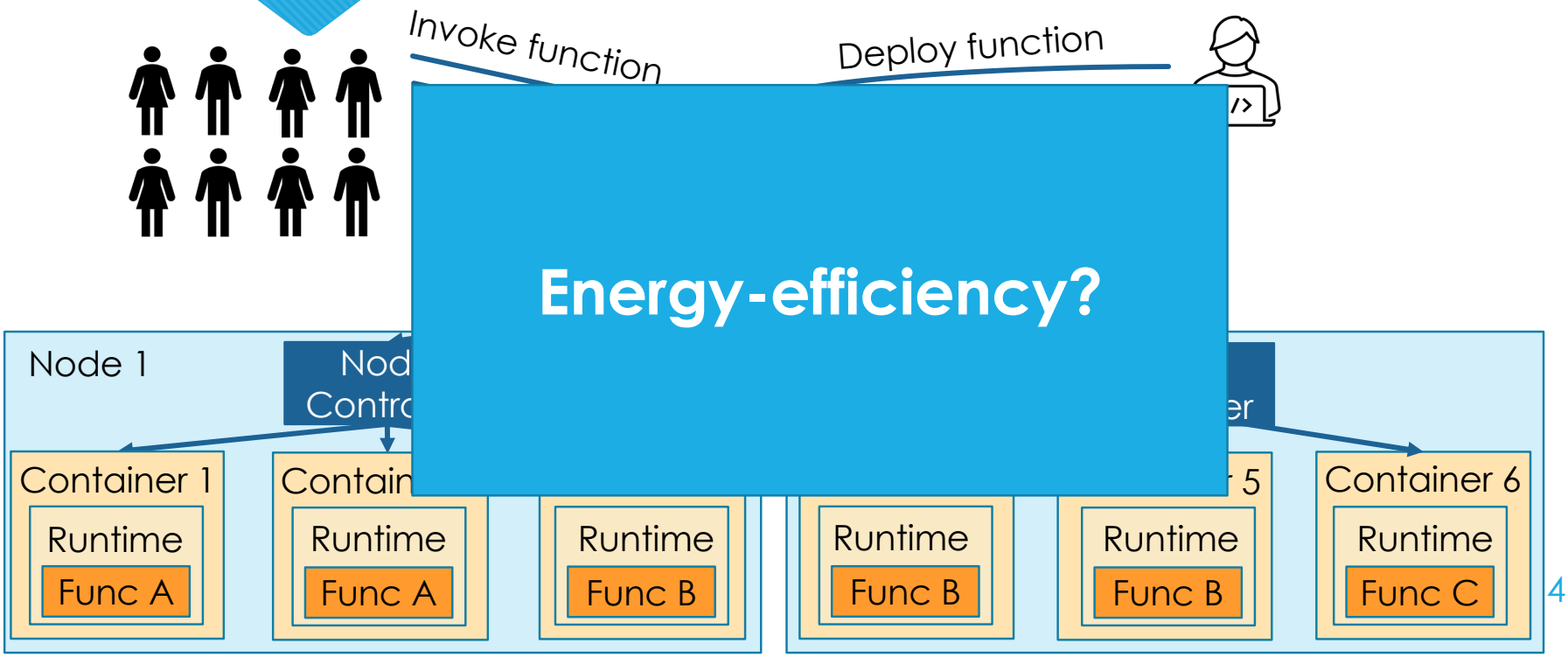
What is serverless computing?

- Serverless computing popular cloud paradigm
  - Users deploy applications, providers provision resources
- Many benefits
  - Simple and modular programming
  - Automatic resource scaling
  - Pay-as-you-go model
- AWS Lambda, Microsoft Azure, IBM Cloud

How serverless computing works?



How serverless computing works?



Contributions

- Characterization of energy-efficiency in serverless environments
- Propose **EcoFaaS**
  - Service Level Objective (SLO) driven energy-efficient serverless system
- Reduces energy consumption by 42%, and tail-latency by 35%

How to Save Energy?

- Obvious approach: voltage-frequency scaling (DVFS)?

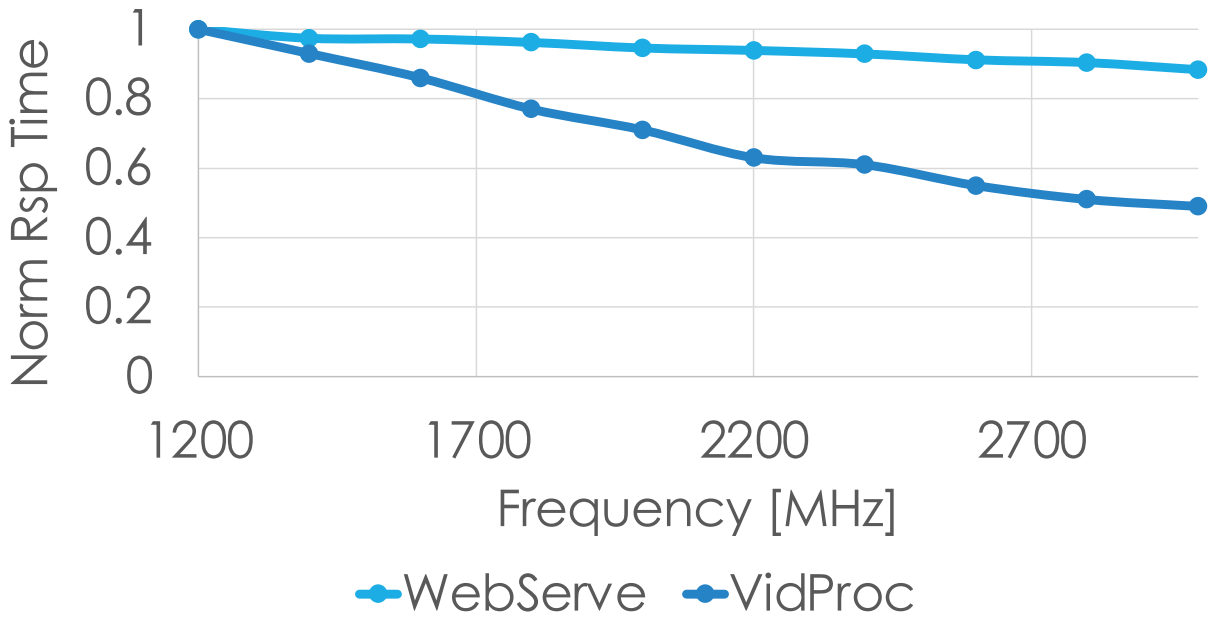
How to Save Energy?

- Obvious app… …FS)?

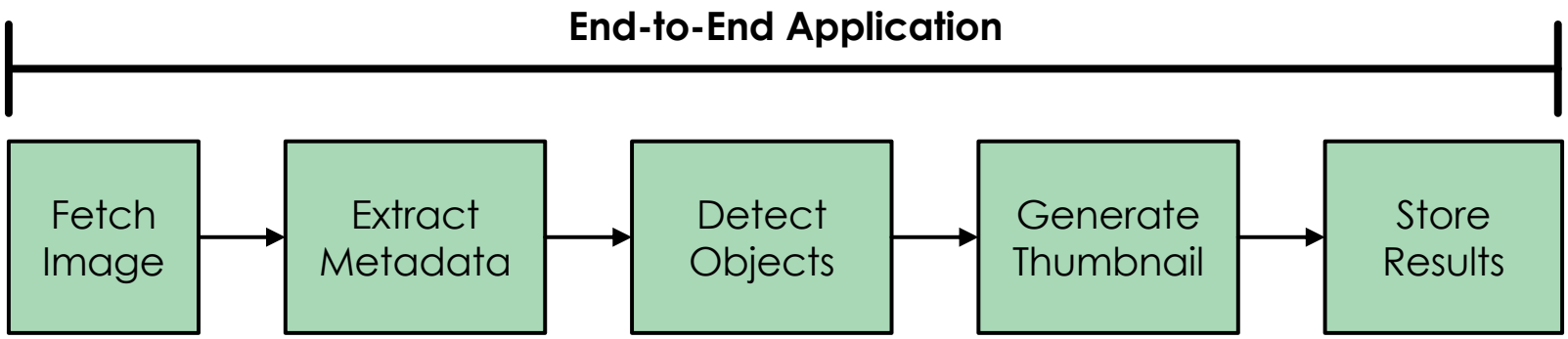
What are the challenges towards energy-efficient serverless systems?

Serverless Functions are Highly Diverse

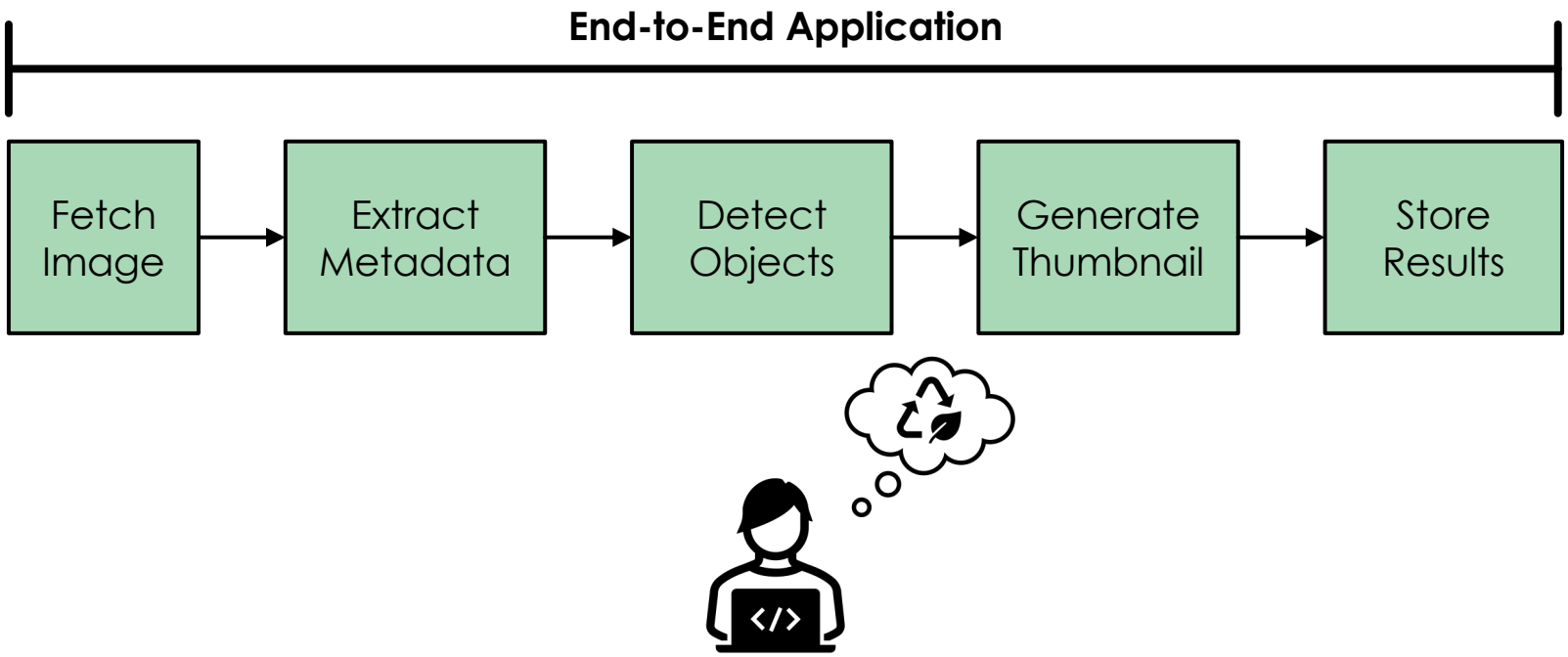
Serverless Functions are Highly Diverse



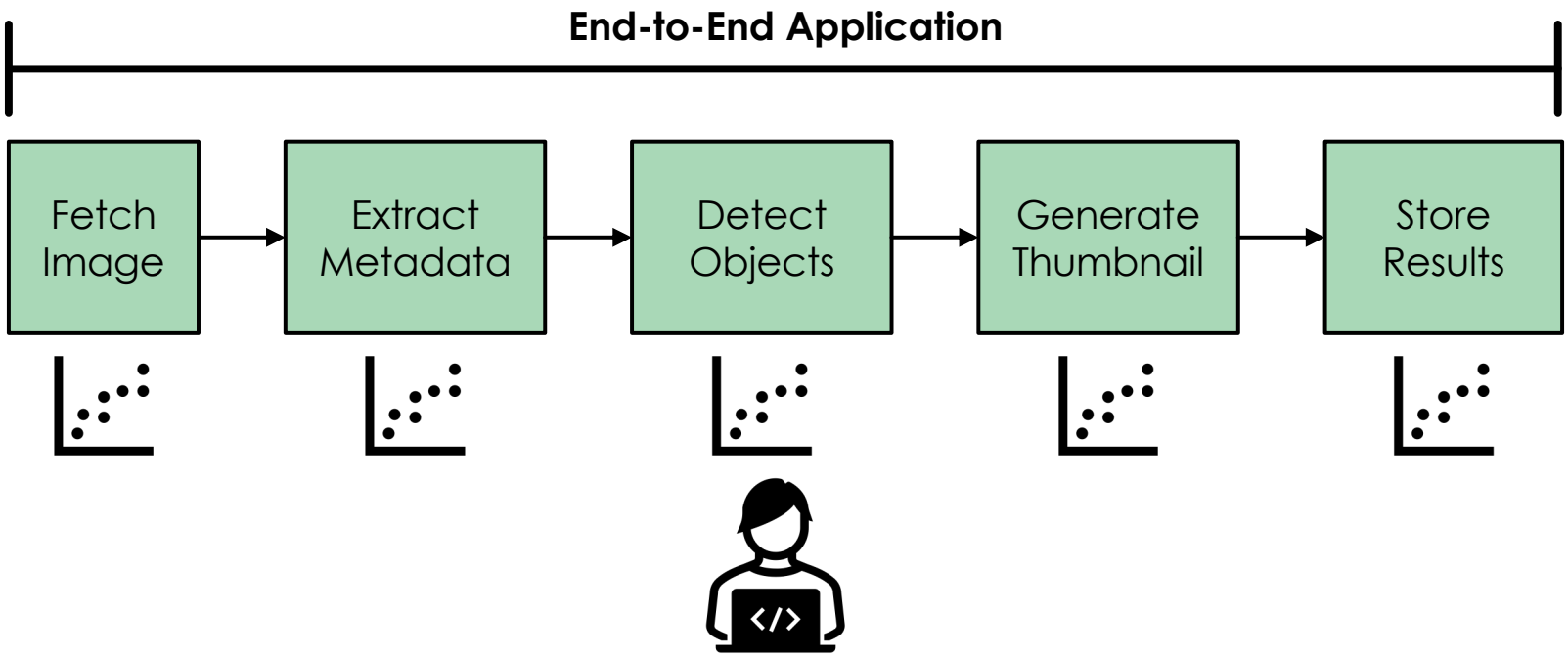
Serverless Functions are Highly Diverse



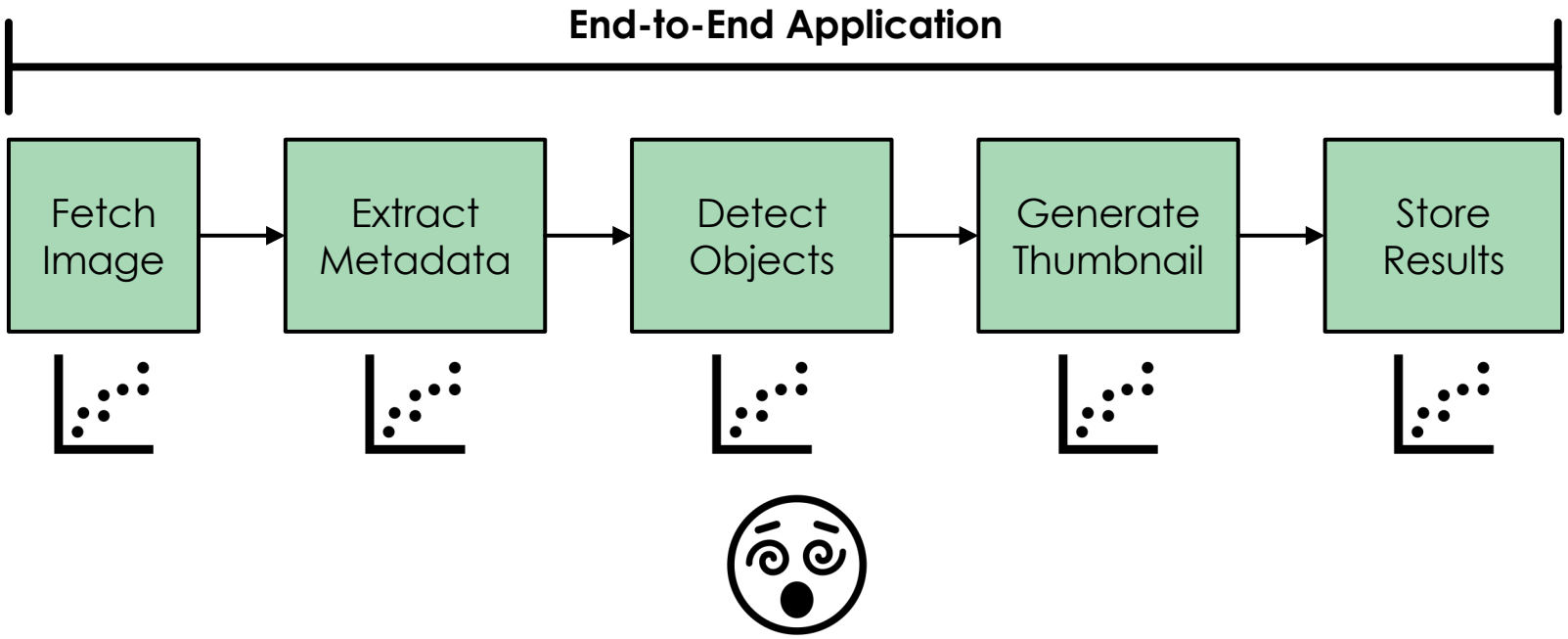
Serverless Functions are Highly Diverse



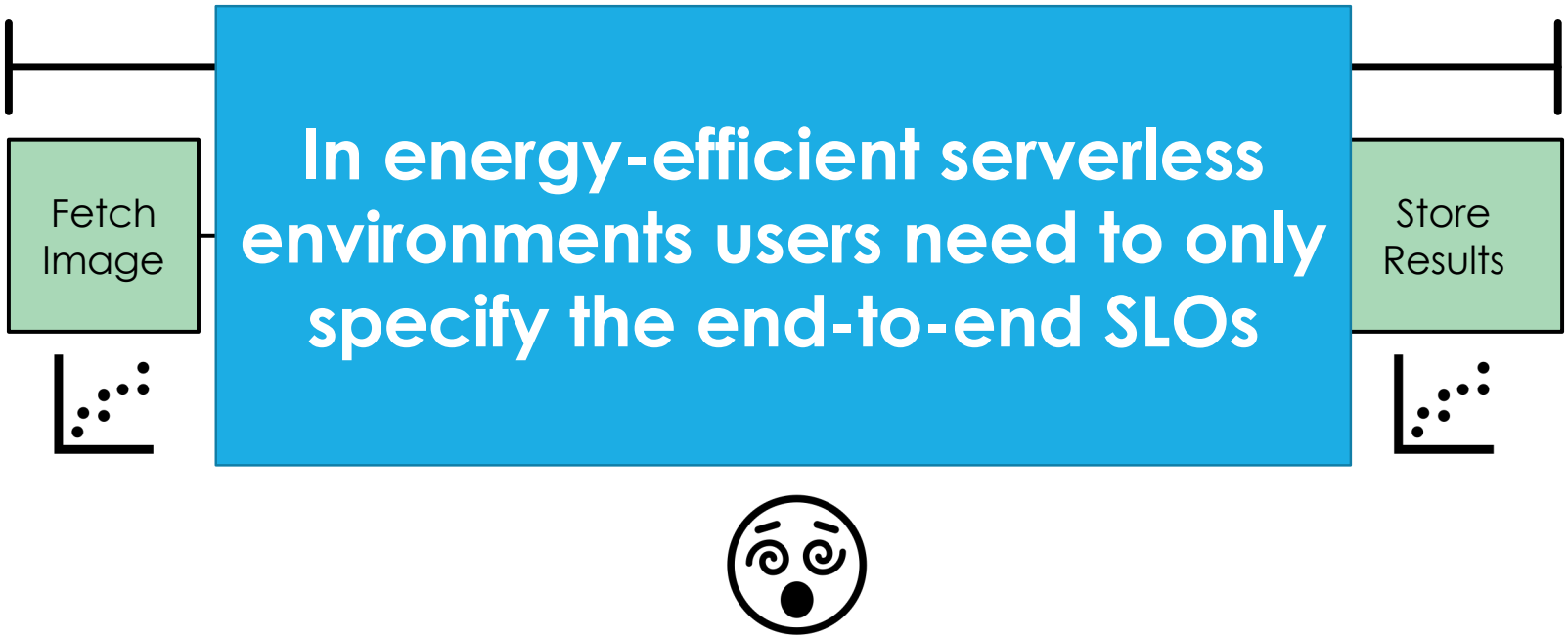
Serverless Functions are Highly Diverse



Serverless Functions are Highly Diverse

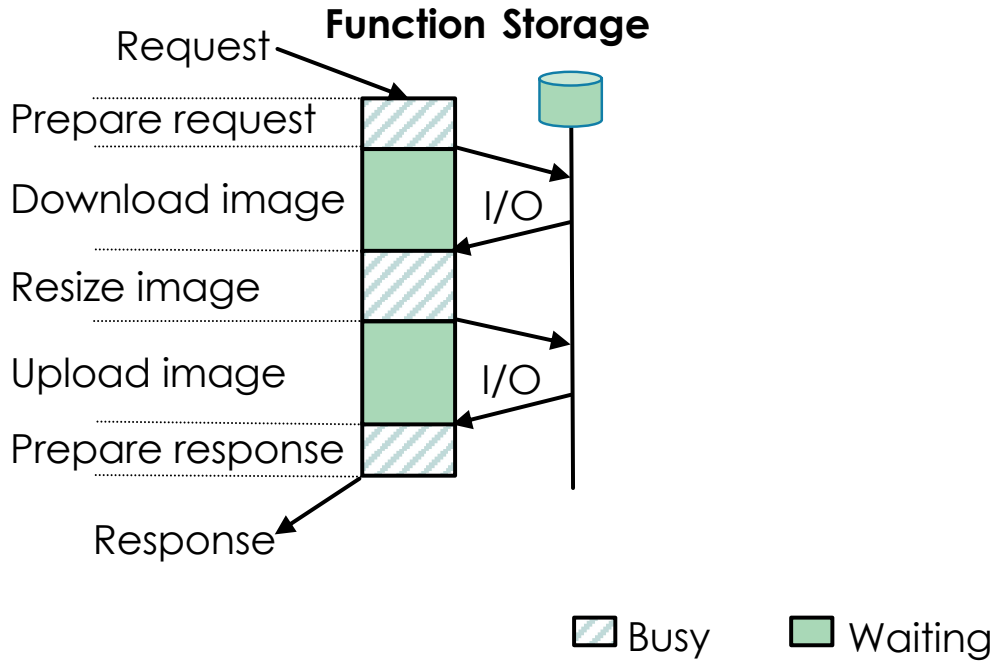


Serverless Functions are Highly Diverse

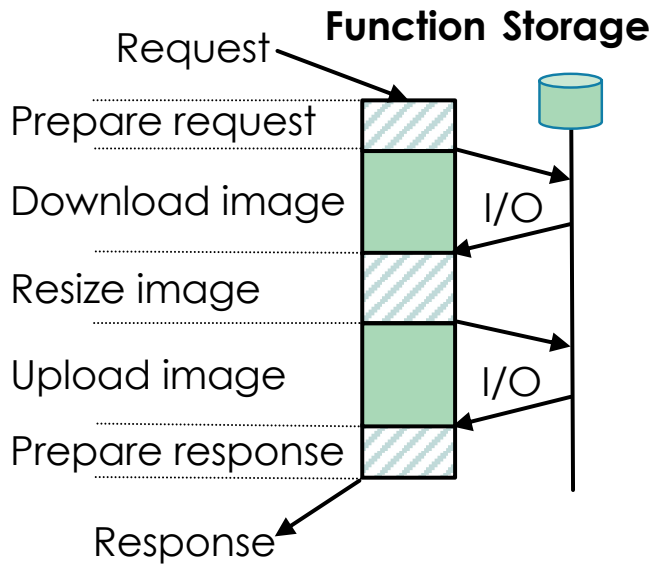


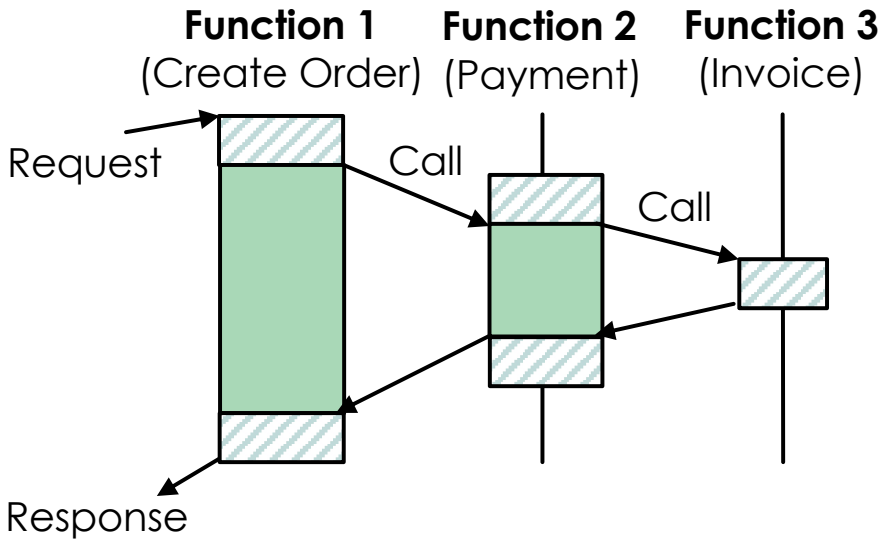
Idle Time Dominates Function Execution

Idle Time Dominates Function Execution



Idle Time Dominates Function Execution





Busy Waiting

Idle Time Dominates Function Execution

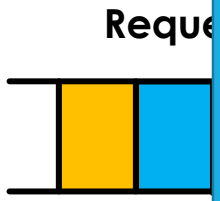
Idle Time Dominates Function Execution

Idle Time Dominates Function Execution

Idle Time Dominates Function Execution

Idle Time Dominates Function Execution

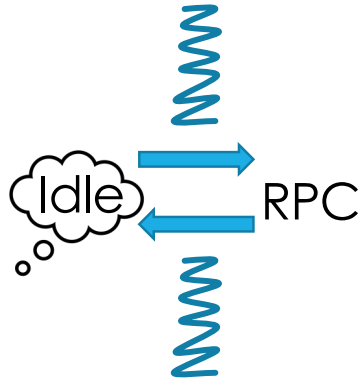
Idle Time Dominates Function Execution



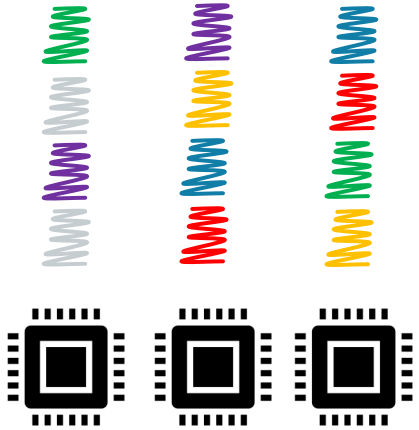
Energy-efficient serverless environments need to exploit the abundant idle time

Frequency Changes are Expensive

Function invocation

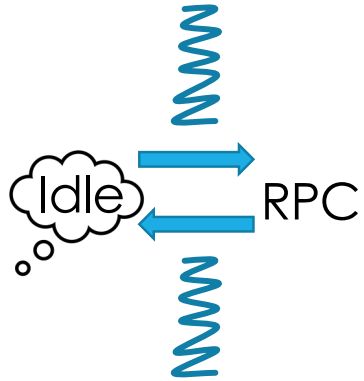


Frequent context switches!

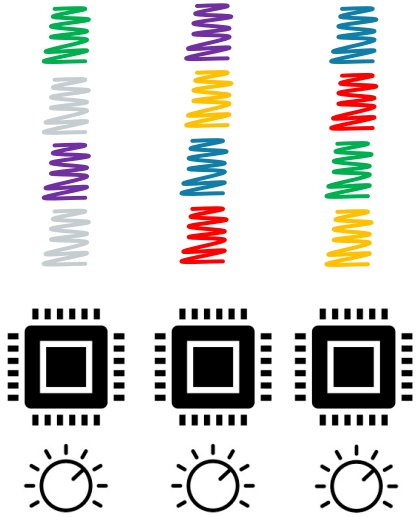


Frequency Changes are Expensive

Function invocation

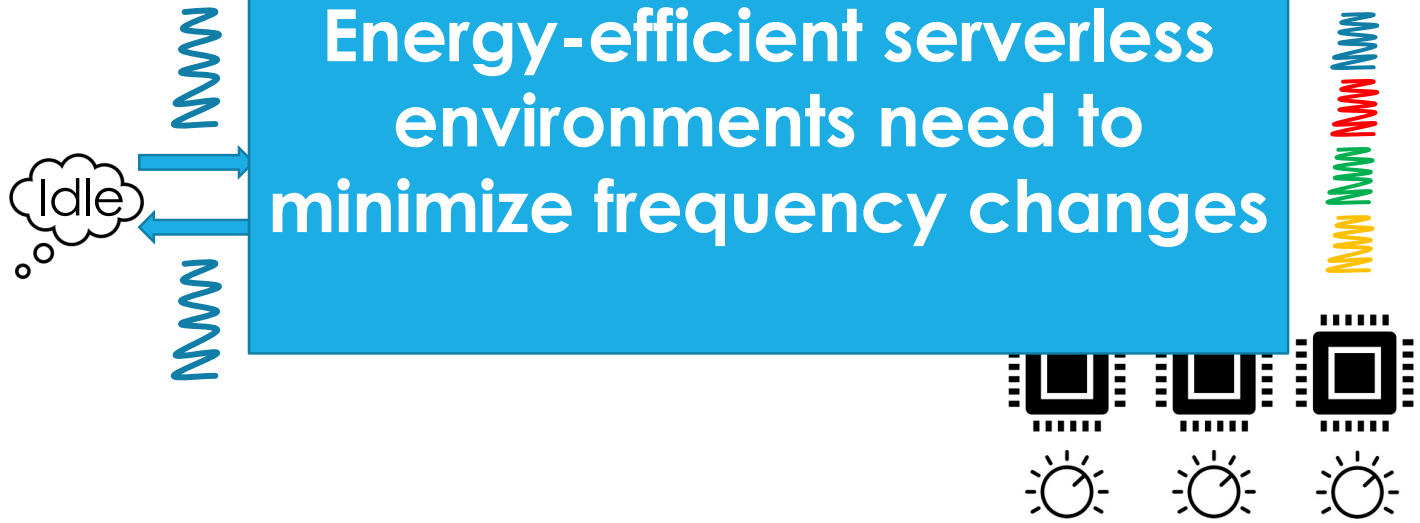


Frequent changes of core frequency!



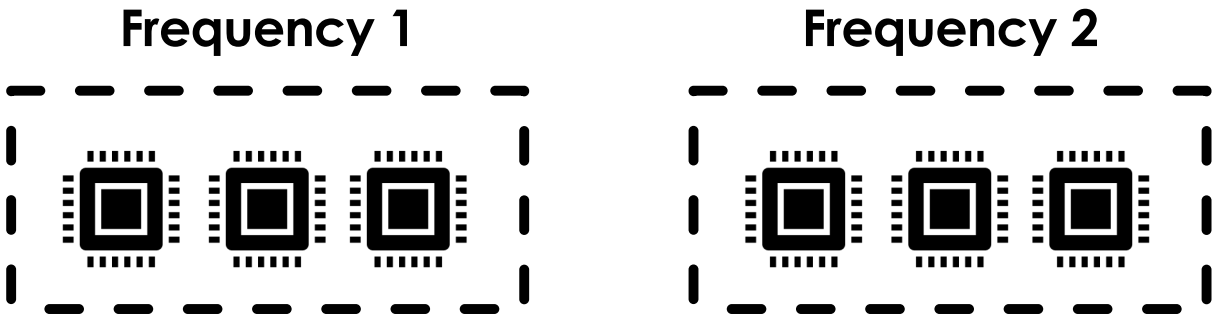
Frequency Changes are Expensive

Function inv f core frequency!

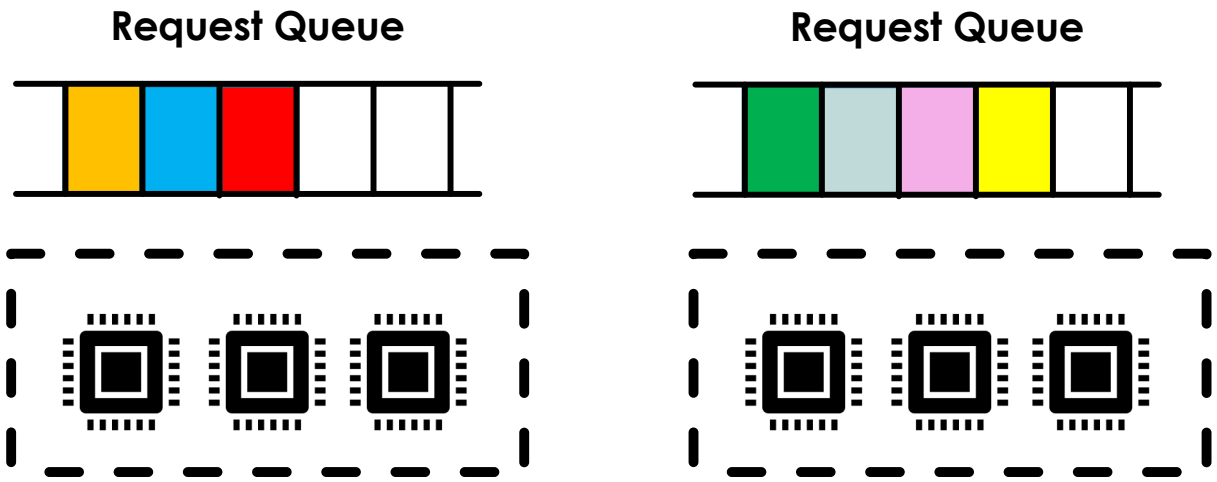


Energy-efficient serverless environments need to minimize frequency changes

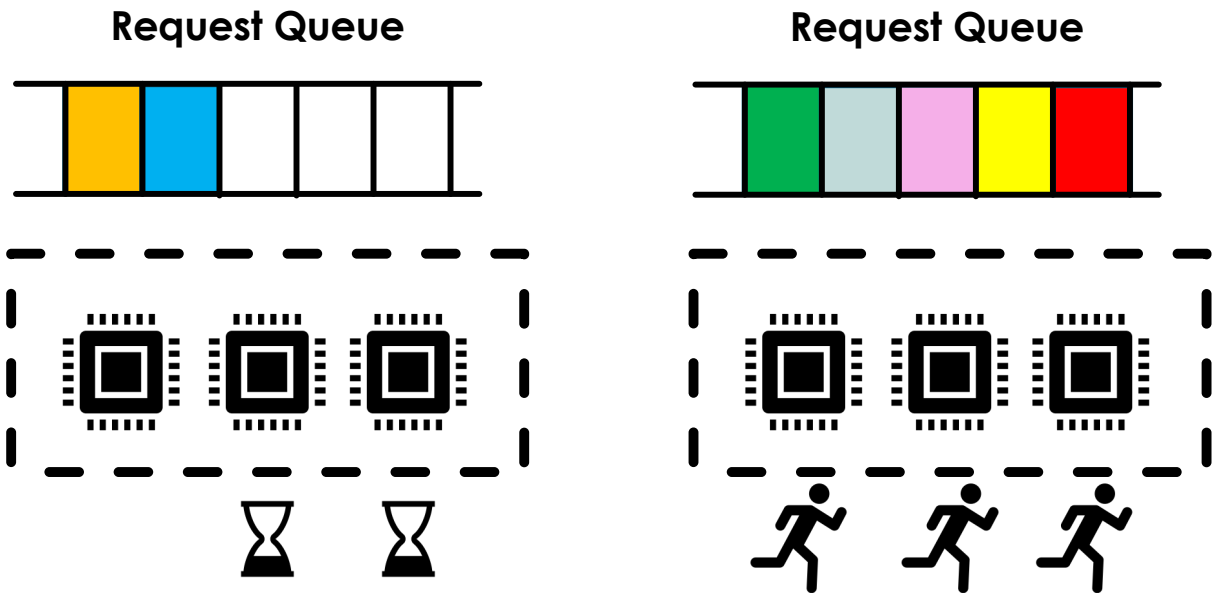
Serverless Workloads are Highly Dynamic



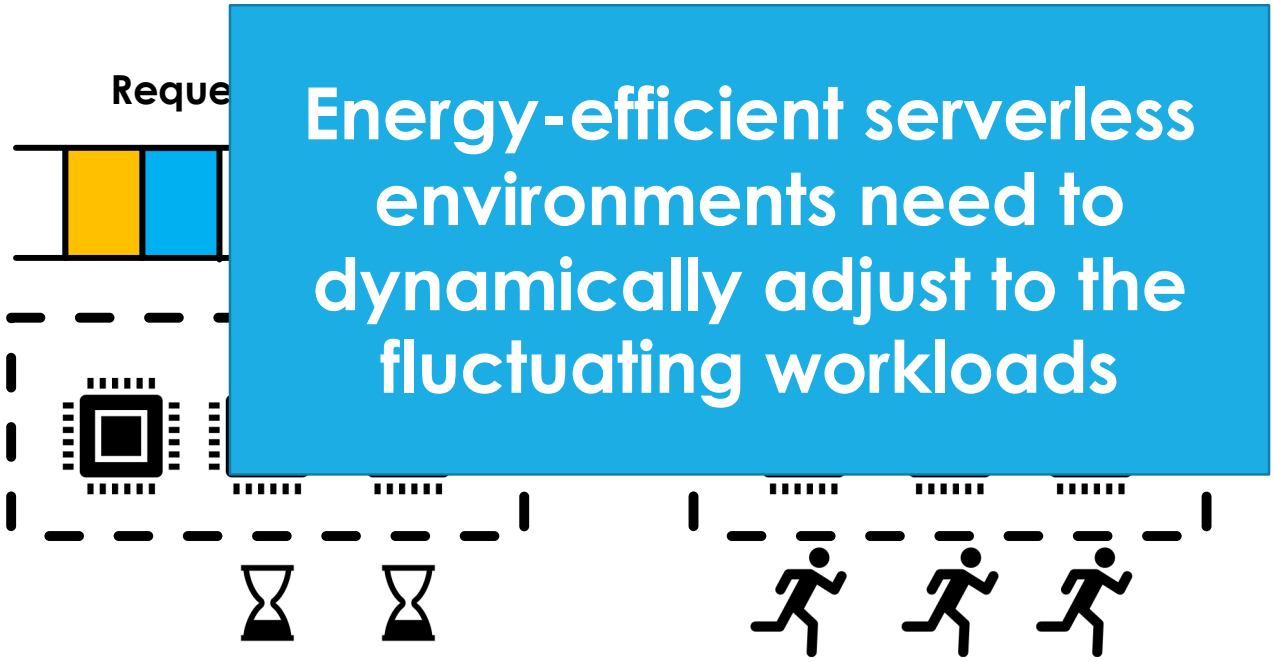
Serverless Workloads are Highly Dynamic



Serverless Workloads are Highly Dynamic

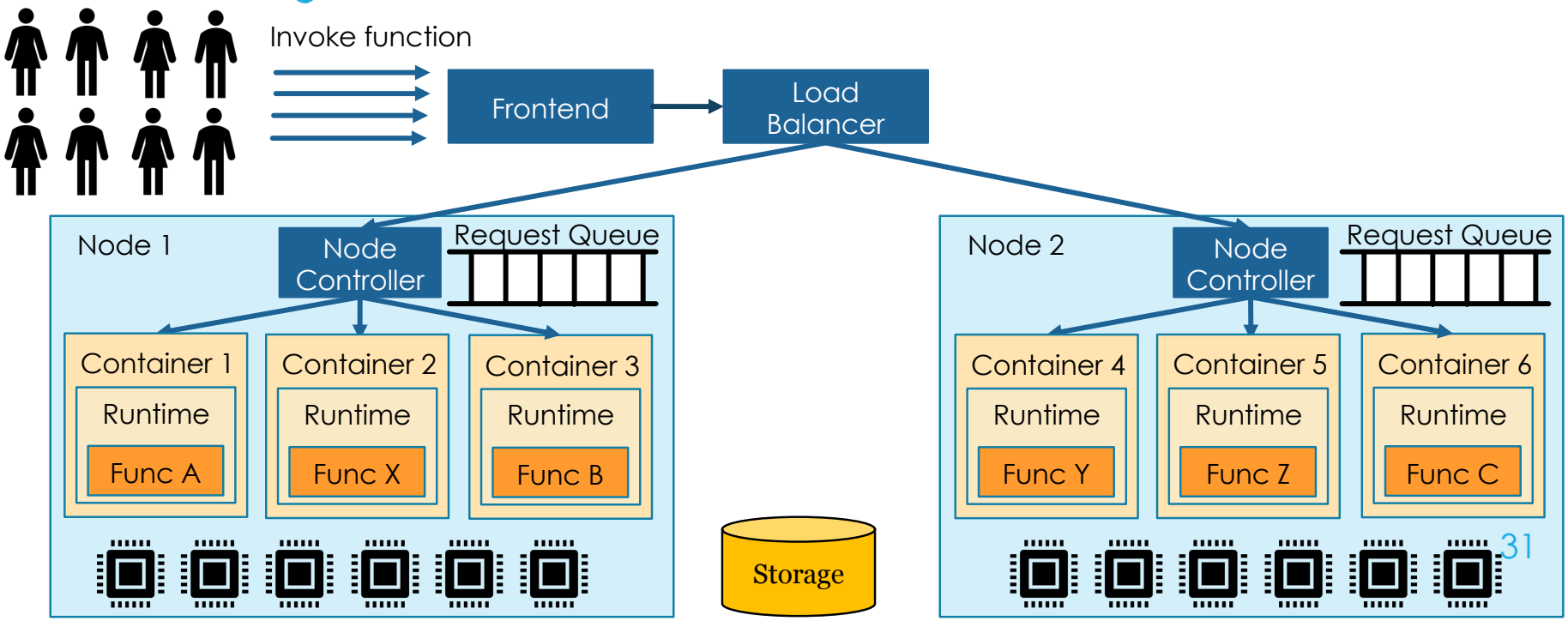


Serverless Workloads are Highly Dynamic

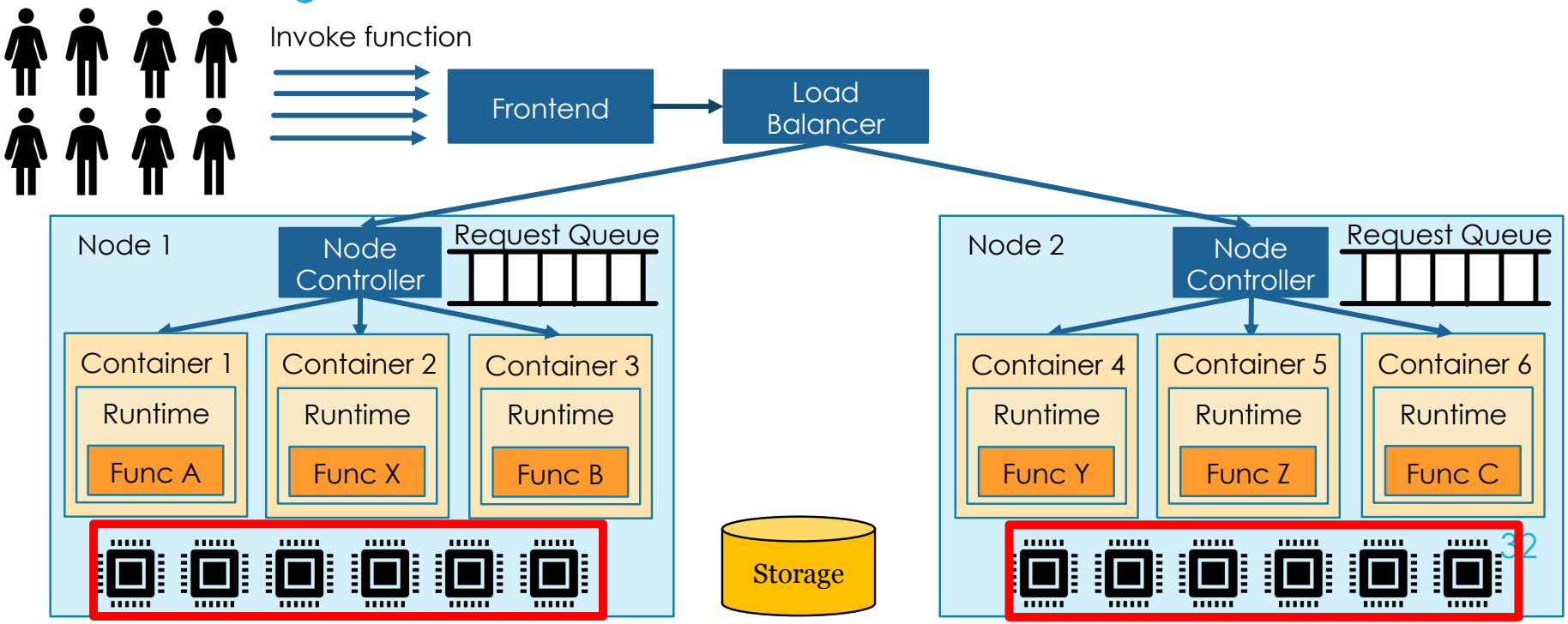


Energy-efficient serverless environments need to dynamically adjust to the fluctuating workloads

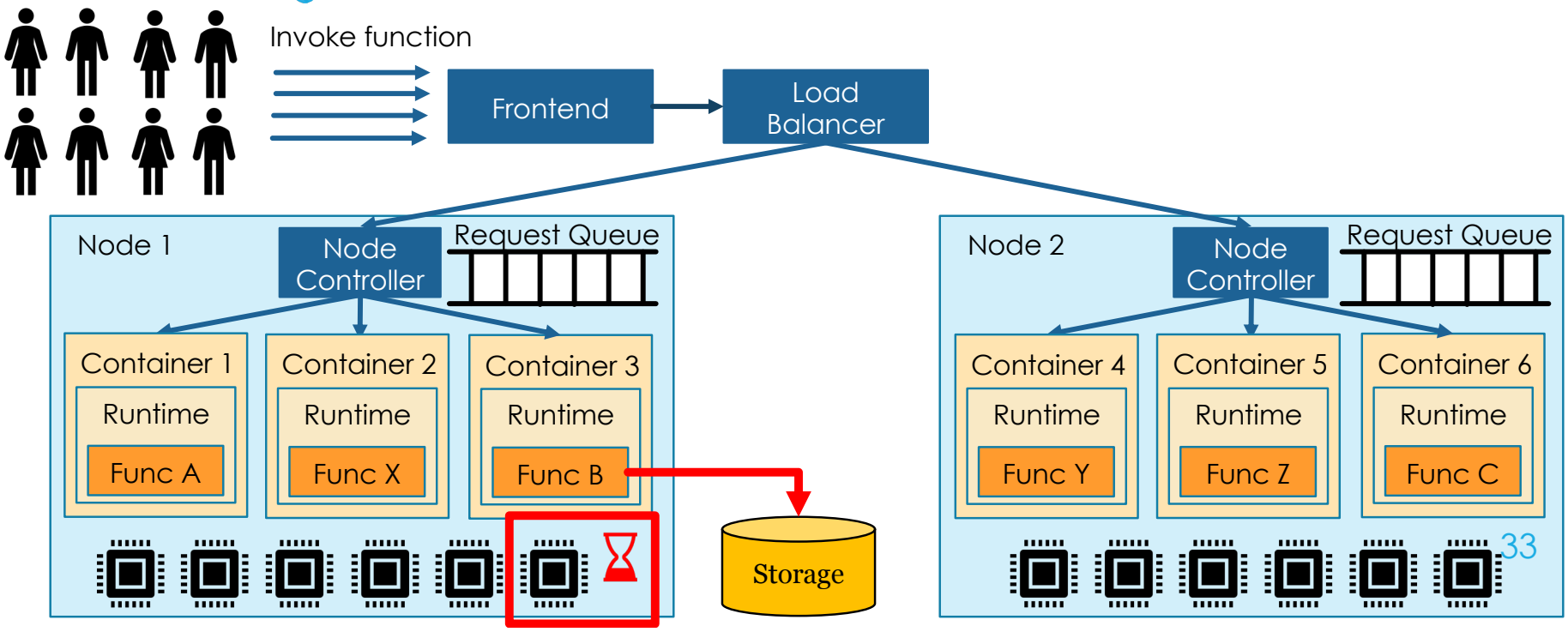
Conventional Systems are not Optimized for Energy-Efficiency



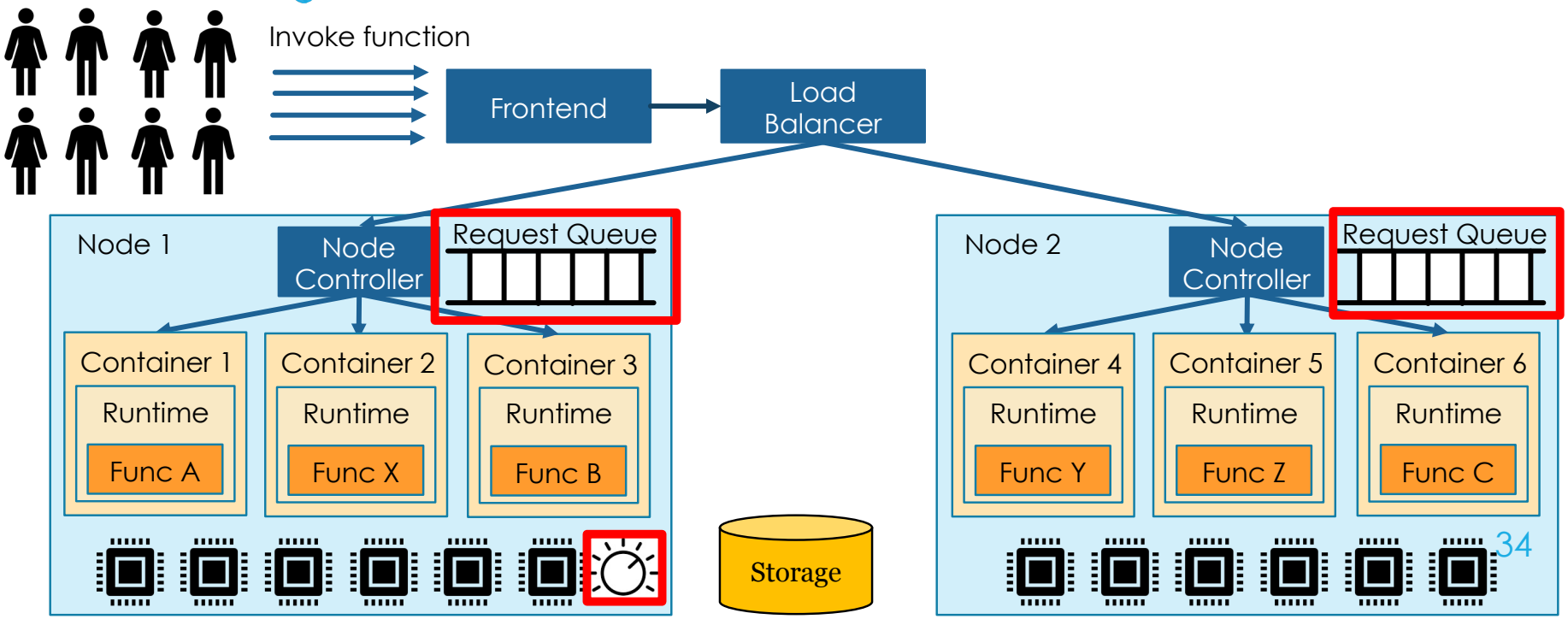
Conventional Systems are not Optimized for Energy-Efficiency



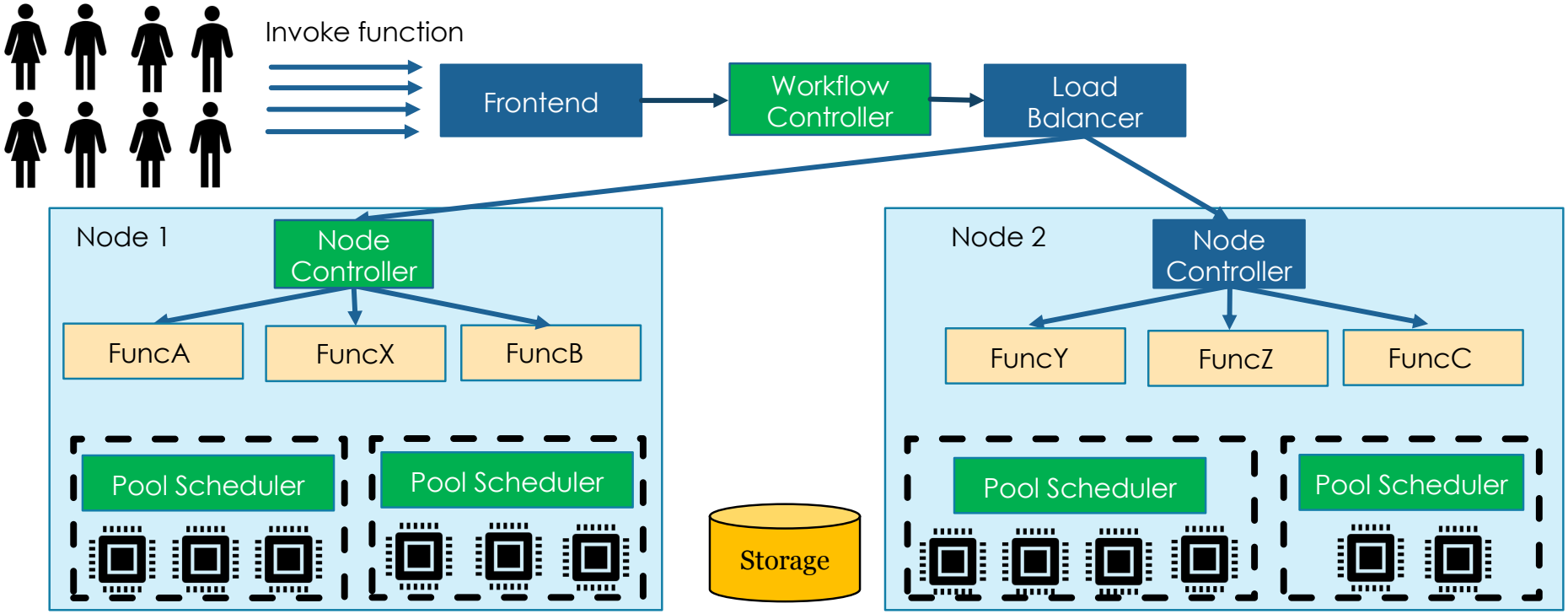
Conventional Systems are not Optimized for Energy-Efficiency



Conventional Systems are not Optimized for Energy-Efficiency



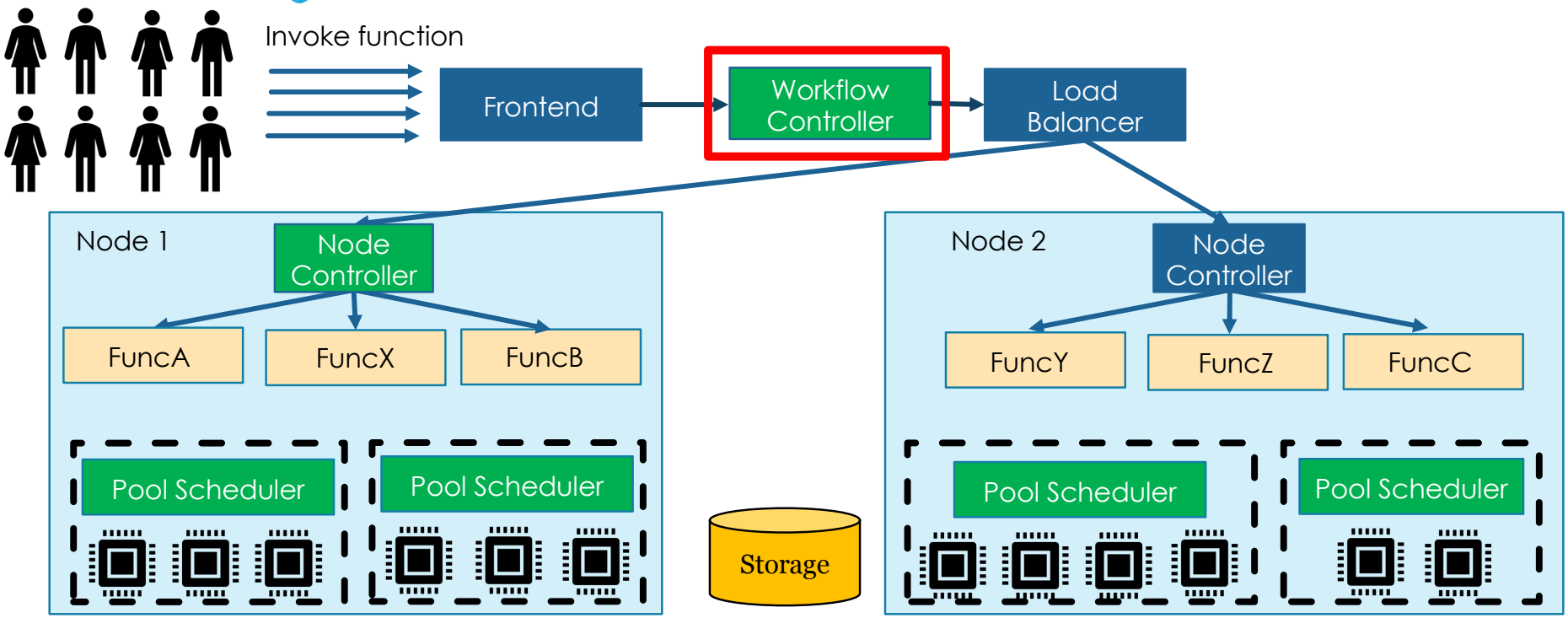
EcoFaaS: An Energy Management Framework for Serverless Environments



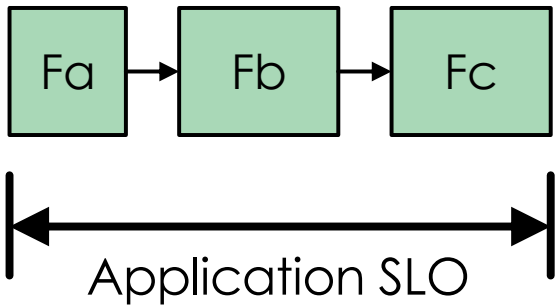
EcoFaaS Key Ideas

1. **EcoFaaS is driven by SLO metrics**

EcoFaaS: An Energy Management Framework for Serverless Environments



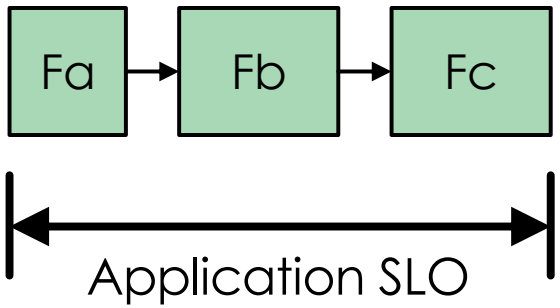
Choosing CPU Frequency with EcoFaaS



Delay-Power Table

	Fa	Fb	Fc
f_1	t_{1a}, E_{1a}	t_{1b}, E_{1b}	t_{1c}, E_{1c}
f_2	t_{2a}, E_{2a}	t_{2b}, E_{2b}	t_{2c}, E_{2c}
f_3	t_{3a}, E_{3a}	t_{3b}, E_{3b}	t_{3c}, E_{3c}

Choosing CPU Frequency with EcoFaaS

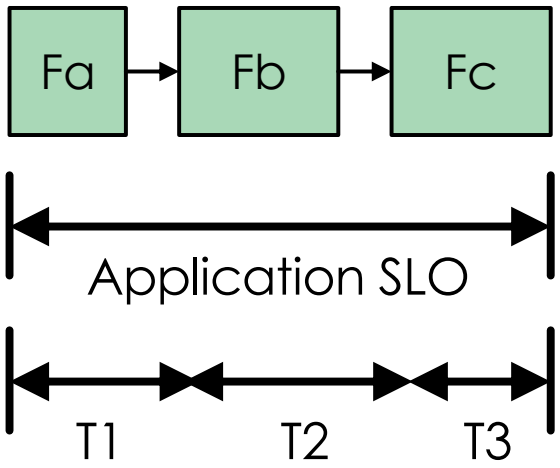


MILP in Workflow Controller

$$\min(\sum Ei)$$

$$s.t.\sum(ti) \leq SLO$$

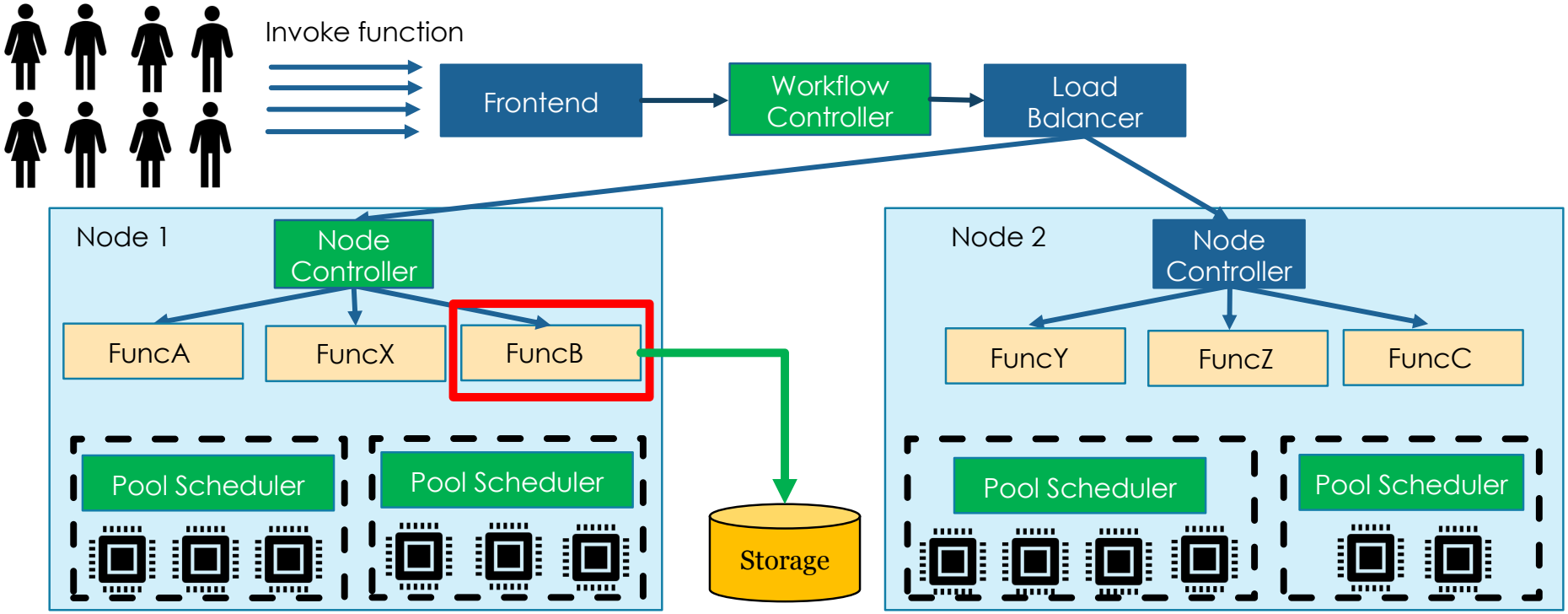
Choosing CPU Frequency with EcoFaaS



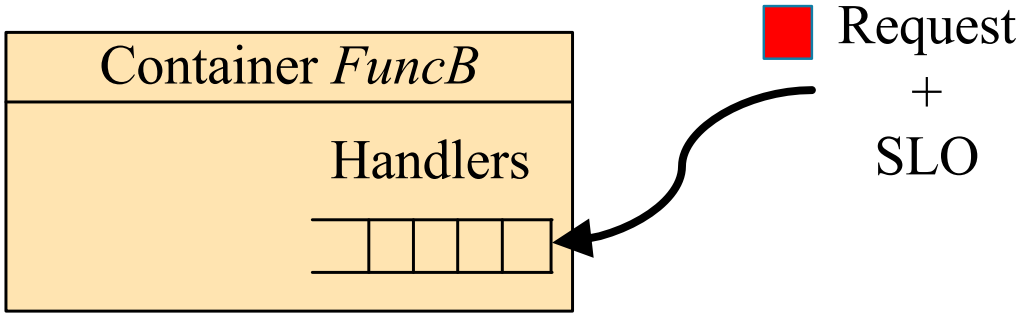
EcoFaaS Key Ideas

1. EcoFaaS is driven by SLO metrics
2. **EcoFaaS profiles and predicts the execution time and energy of function invocations**

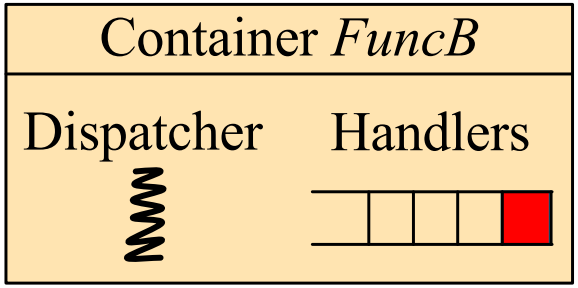
EcoFaaS: An Energy Management Framework for Serverless Environments



Scheduling Requests with EcoFaaS

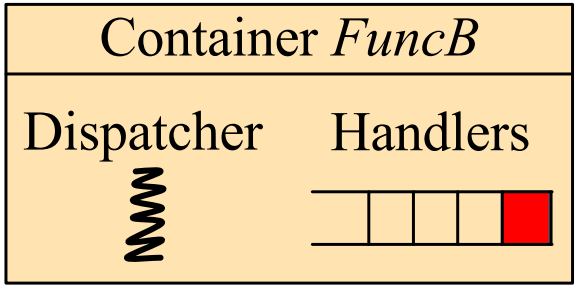


Scheduling Requests with EcoFaaS

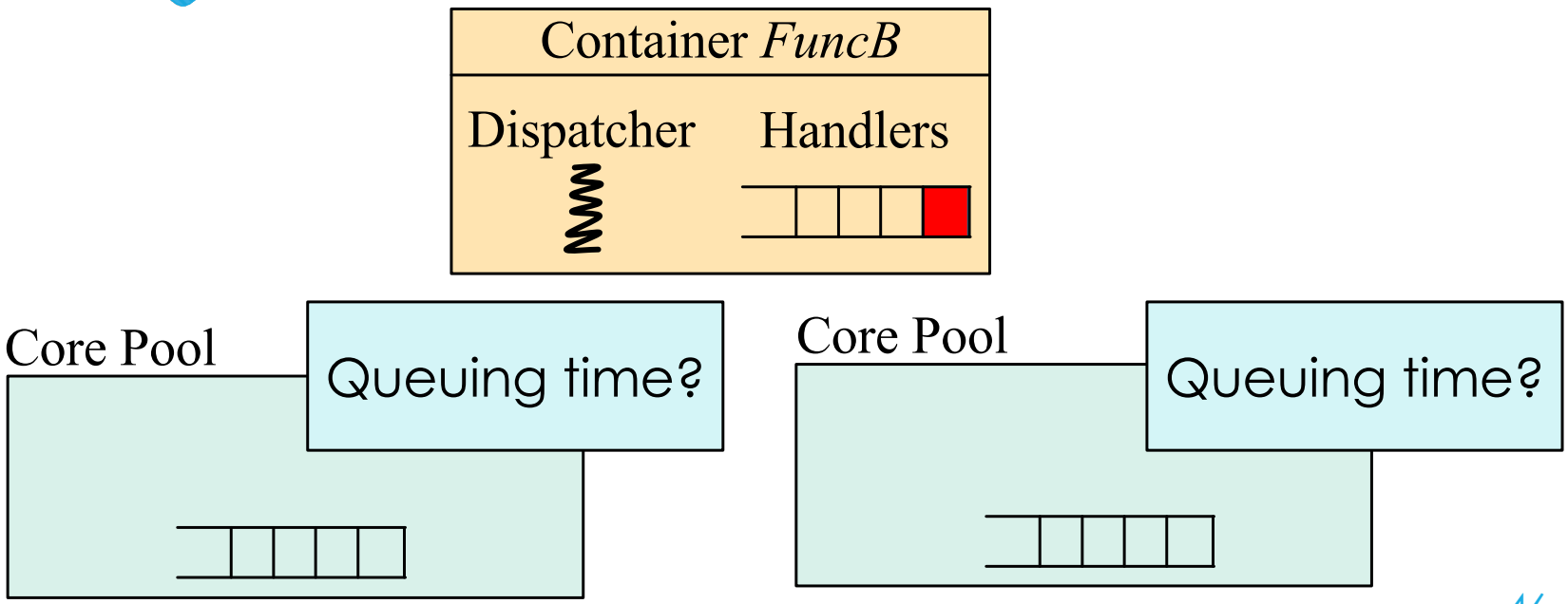


Scheduling Requests with EcoFaaS

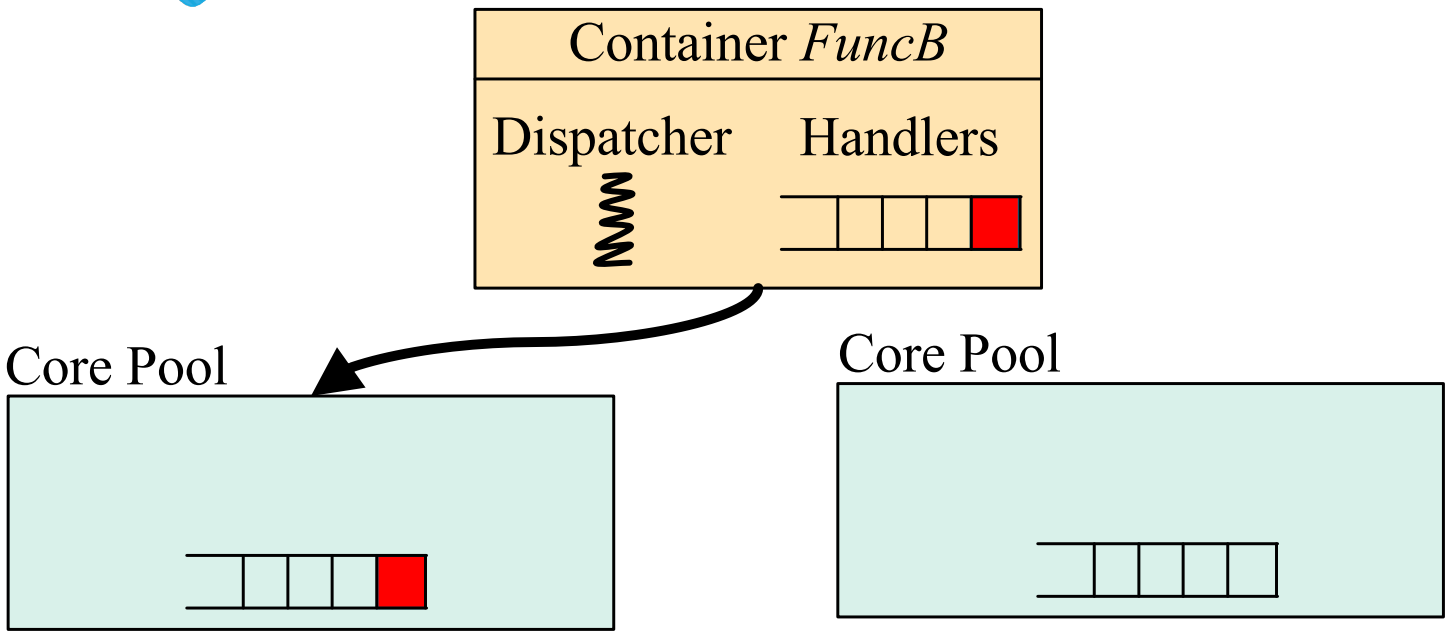
Execution time?
Idle time?
Energy?



Scheduling Requests with EcoFaaS



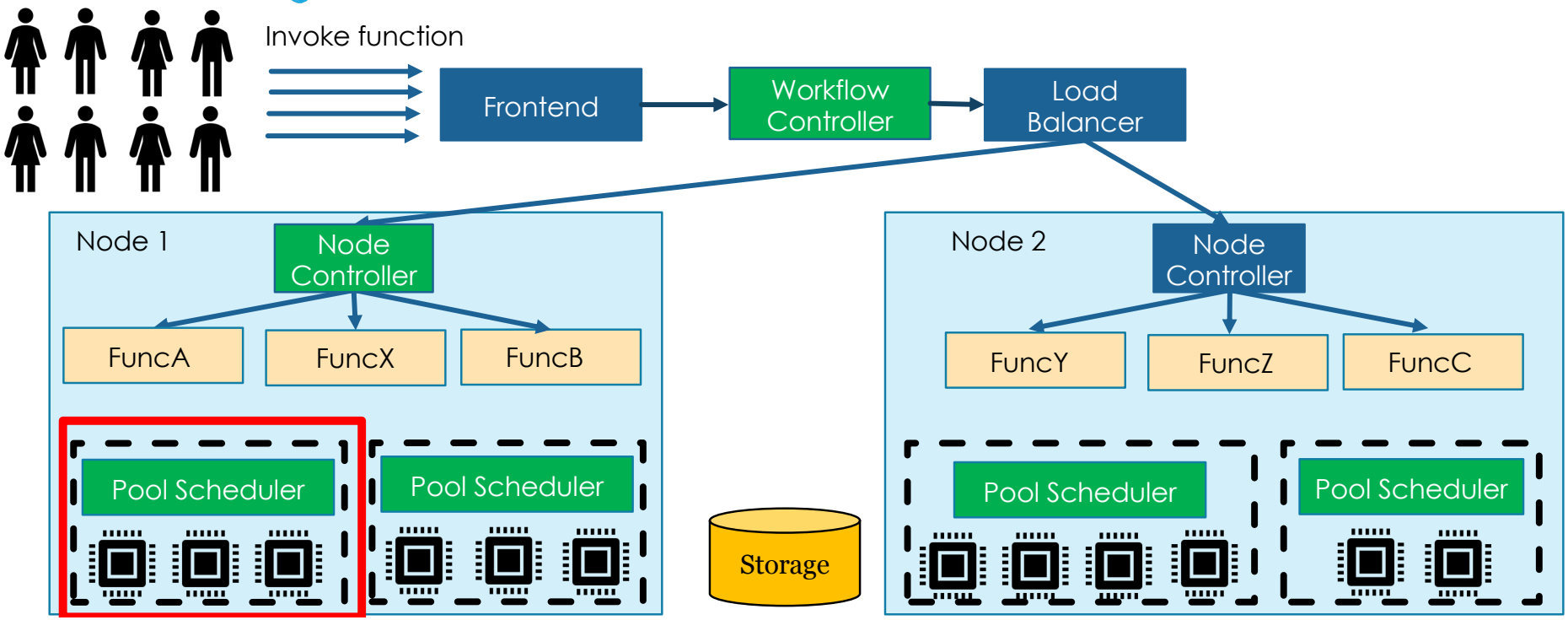
Scheduling Requests with EcoFaaS



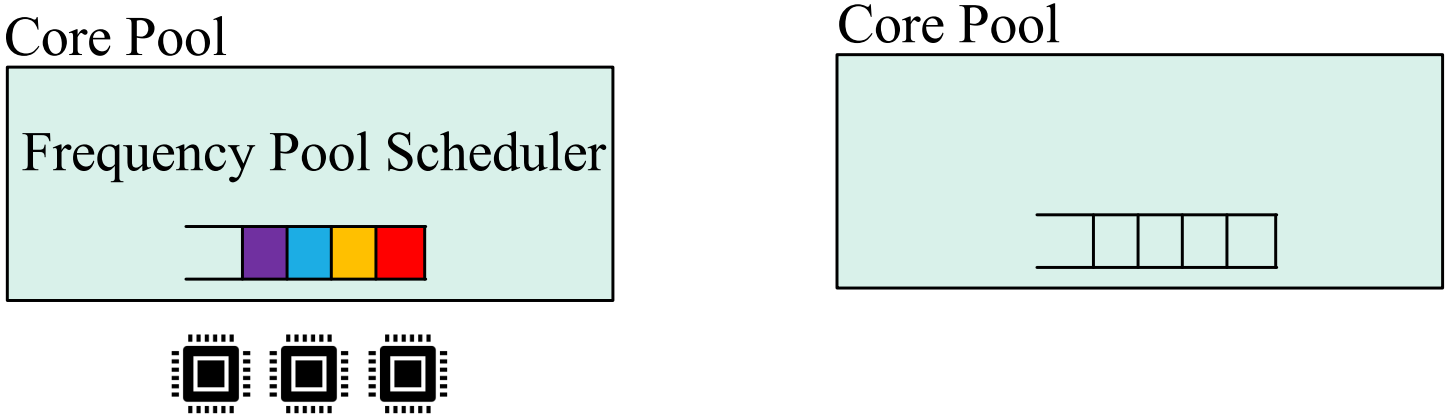
EcoFaaS Key Ideas

1. EcoFaaS is driven by SLO metrics
2. EcoFaaS profiles and predicts the execution time and energy of function invocations
3. **EcoFaaS splits cores into frequency classes**

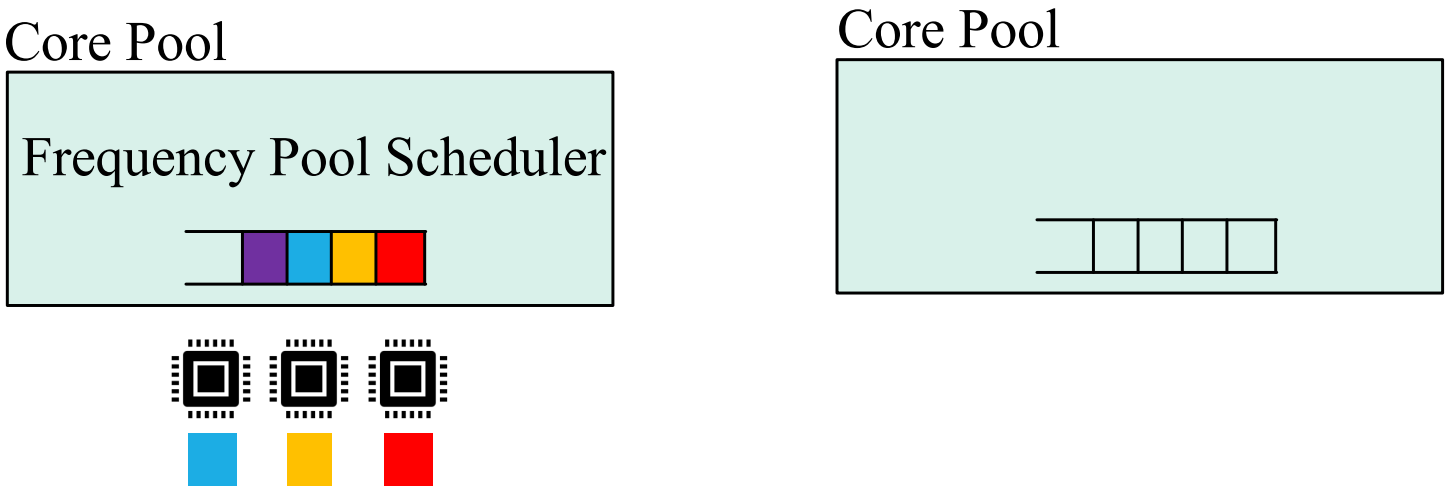
EcoFaaS: An Energy Management Framework for Serverless Environments



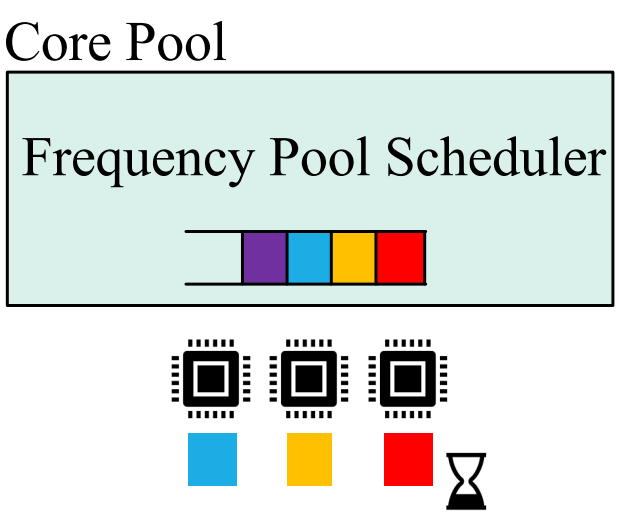
Scheduling Requests with EcoFaaS

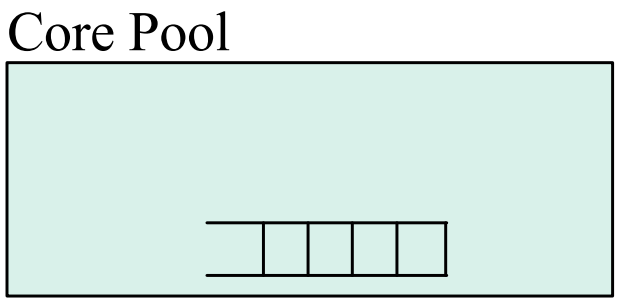


Scheduling Requests with EcoFaaS

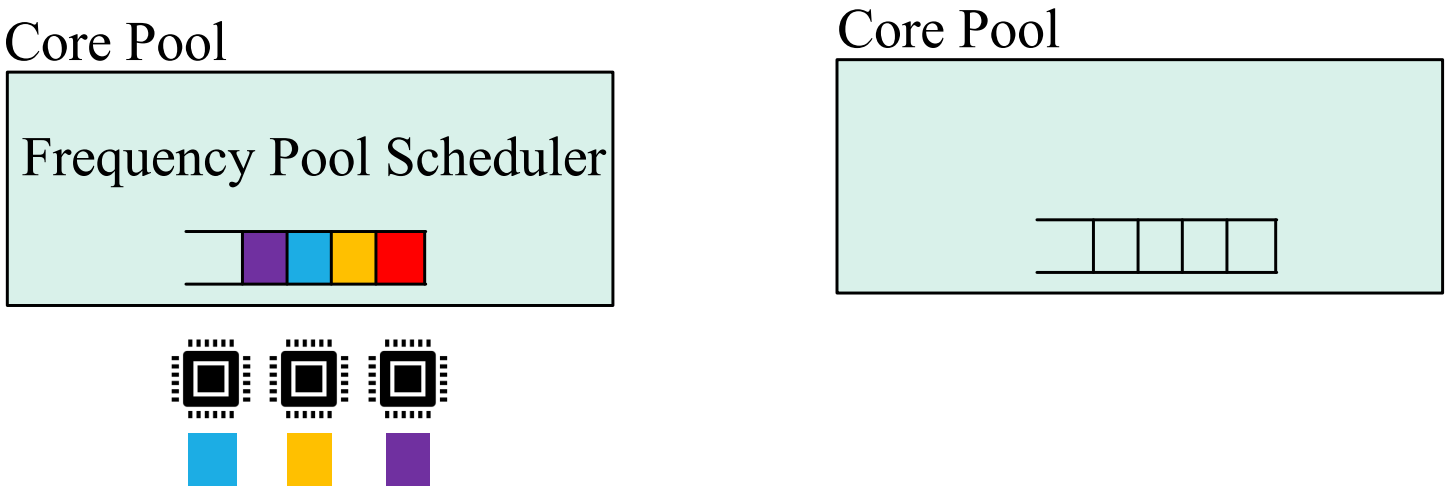


Scheduling Requests with EcoFaaS

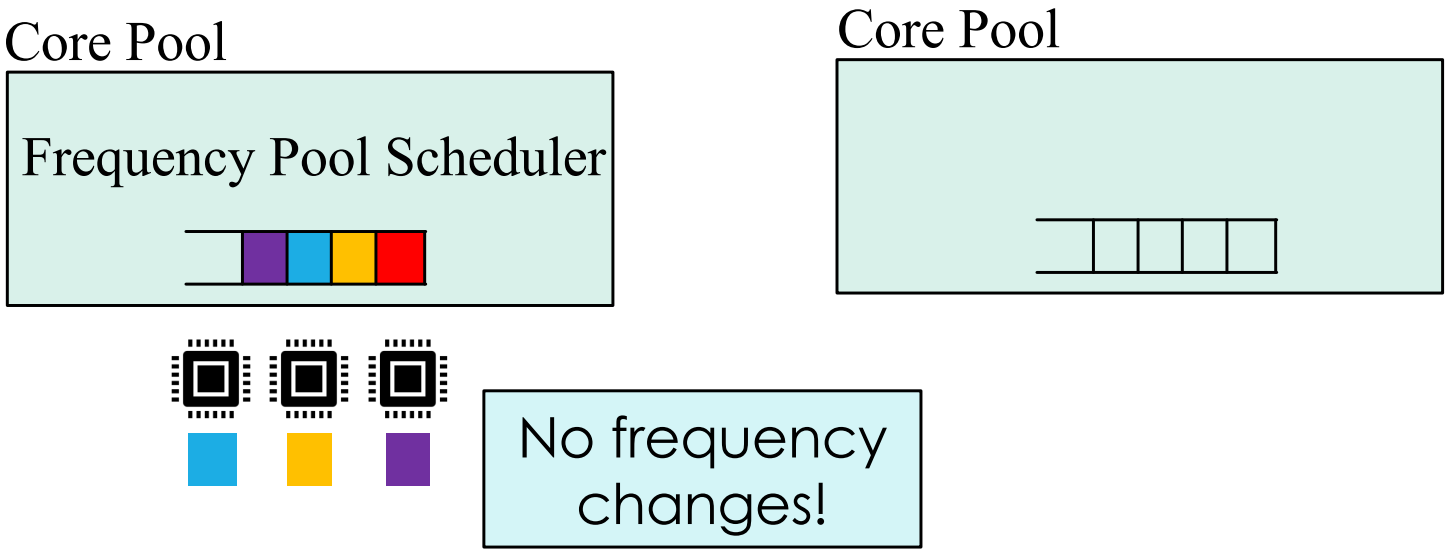




Scheduling Requests with EcoFaaS



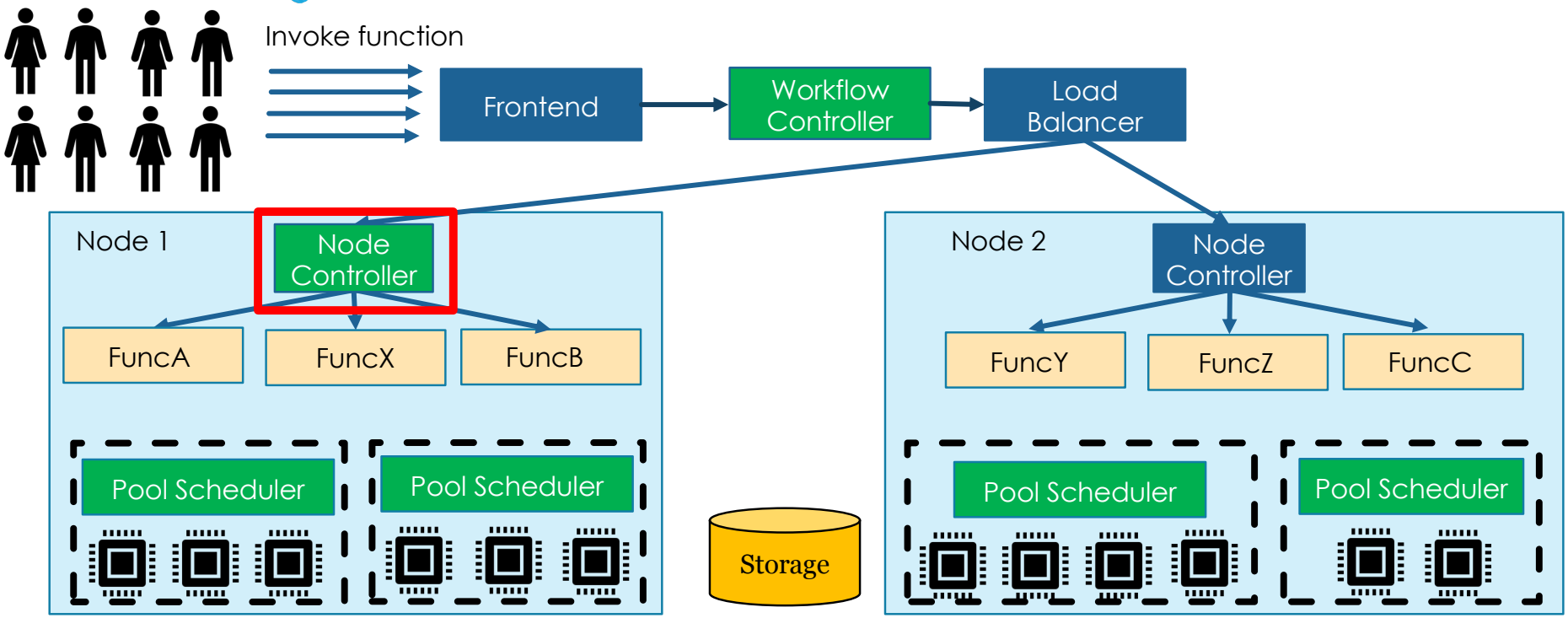
Scheduling Requests with EcoFaaS



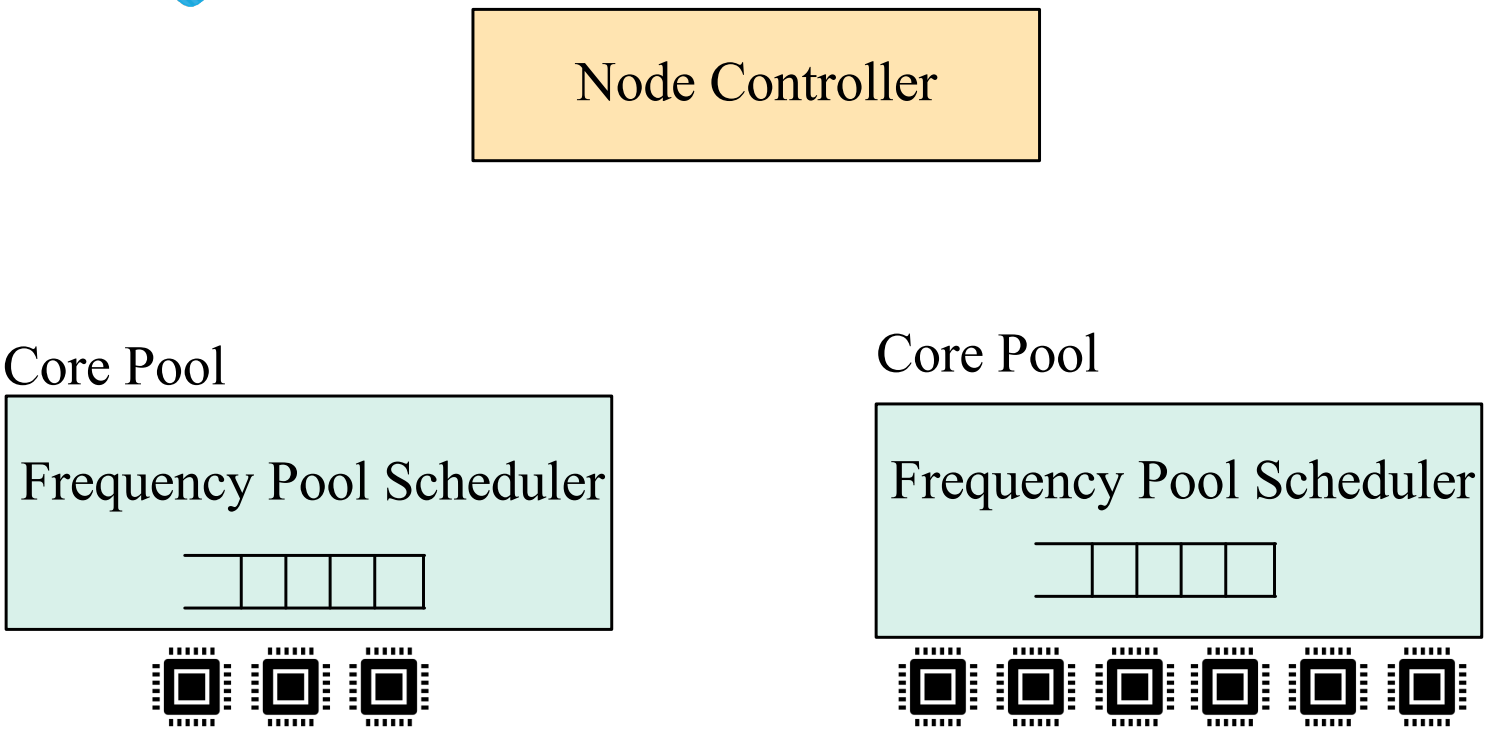
EcoFaaS Key Ideas

1. EcoFaaS is driven by SLO metrics
2. EcoFaaS profiles and predicts the execution time and energy of function invocations
3. EcoFaaS splits cores into frequency classes
4. **EcoFaaS changes pools and pool frequencies dynamically**

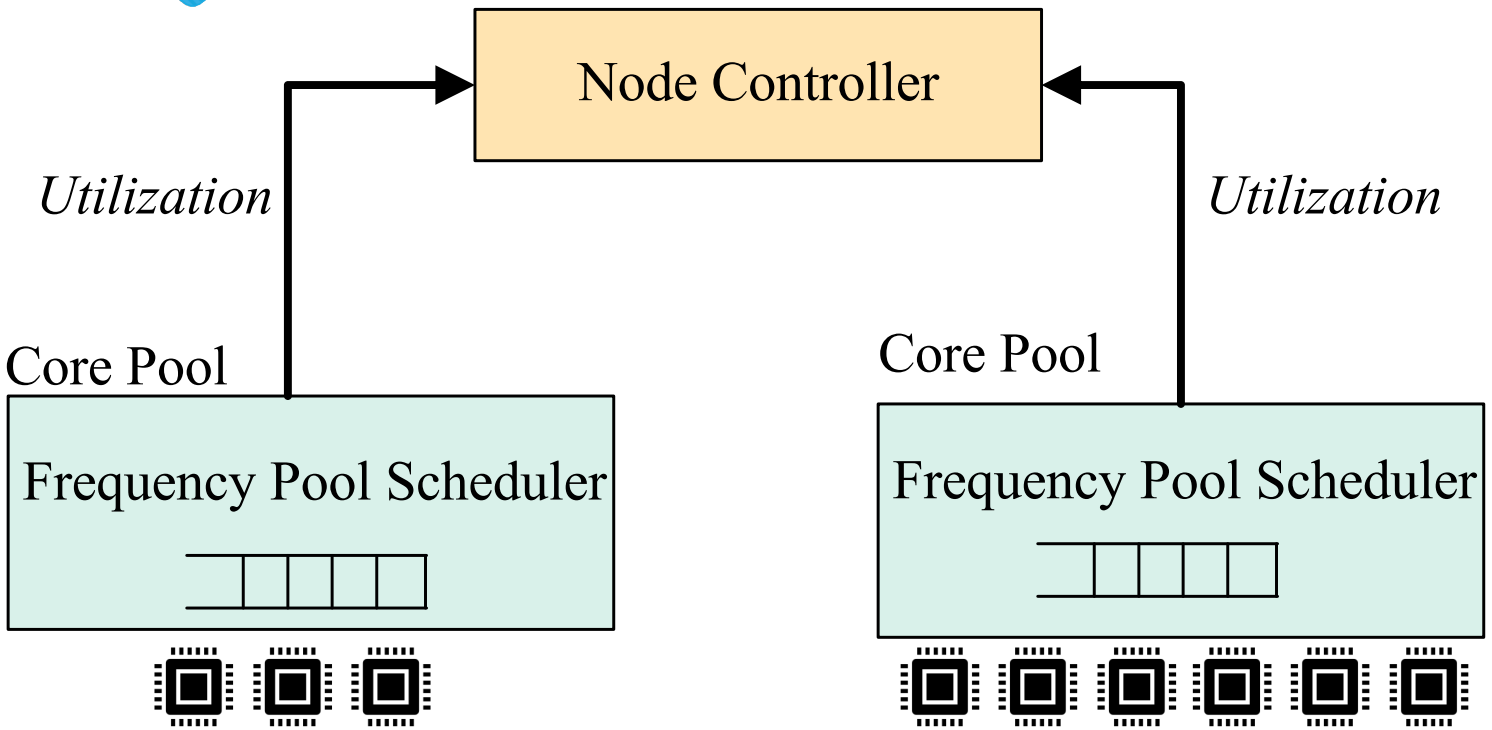
EcoFaaS: An Energy Management Framework for Serverless Environments



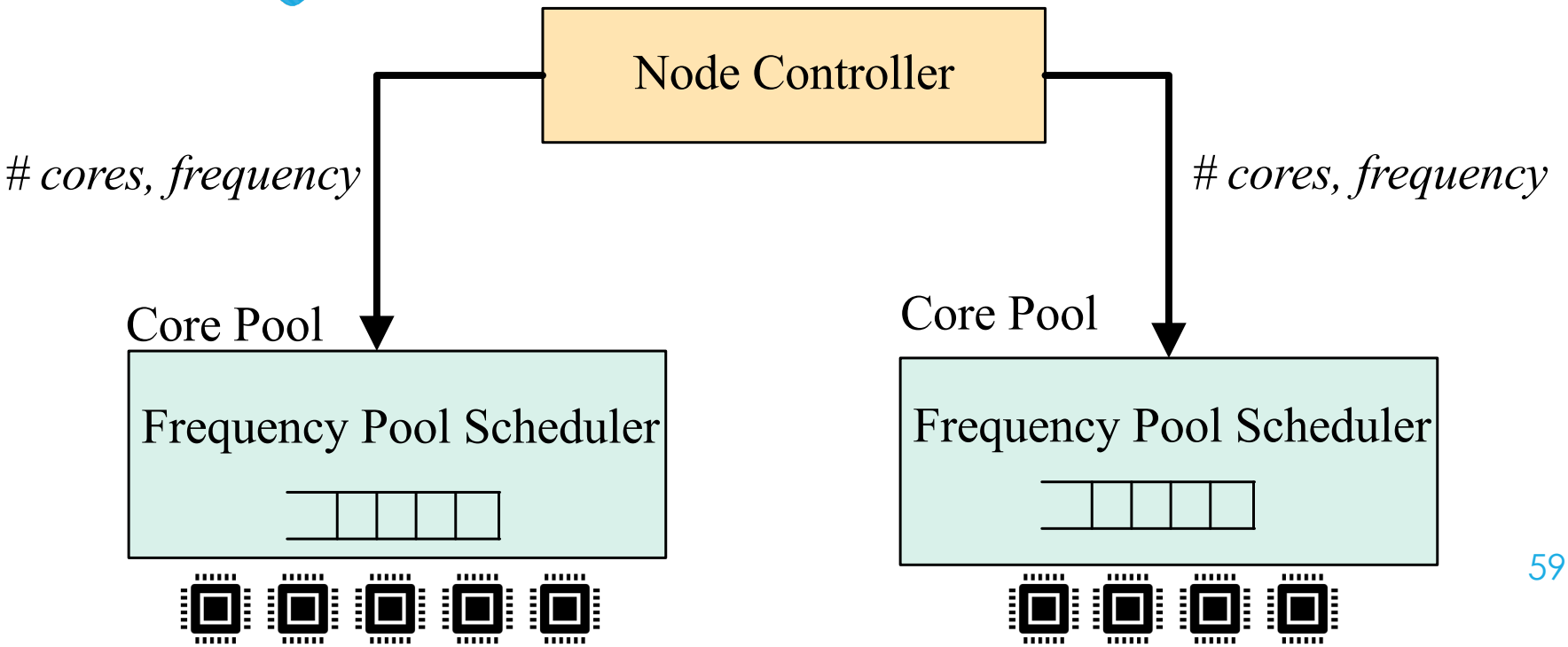
Re-configuring the Pools with EcoFaaS



Re-configuring the Pools with EcoFaaS



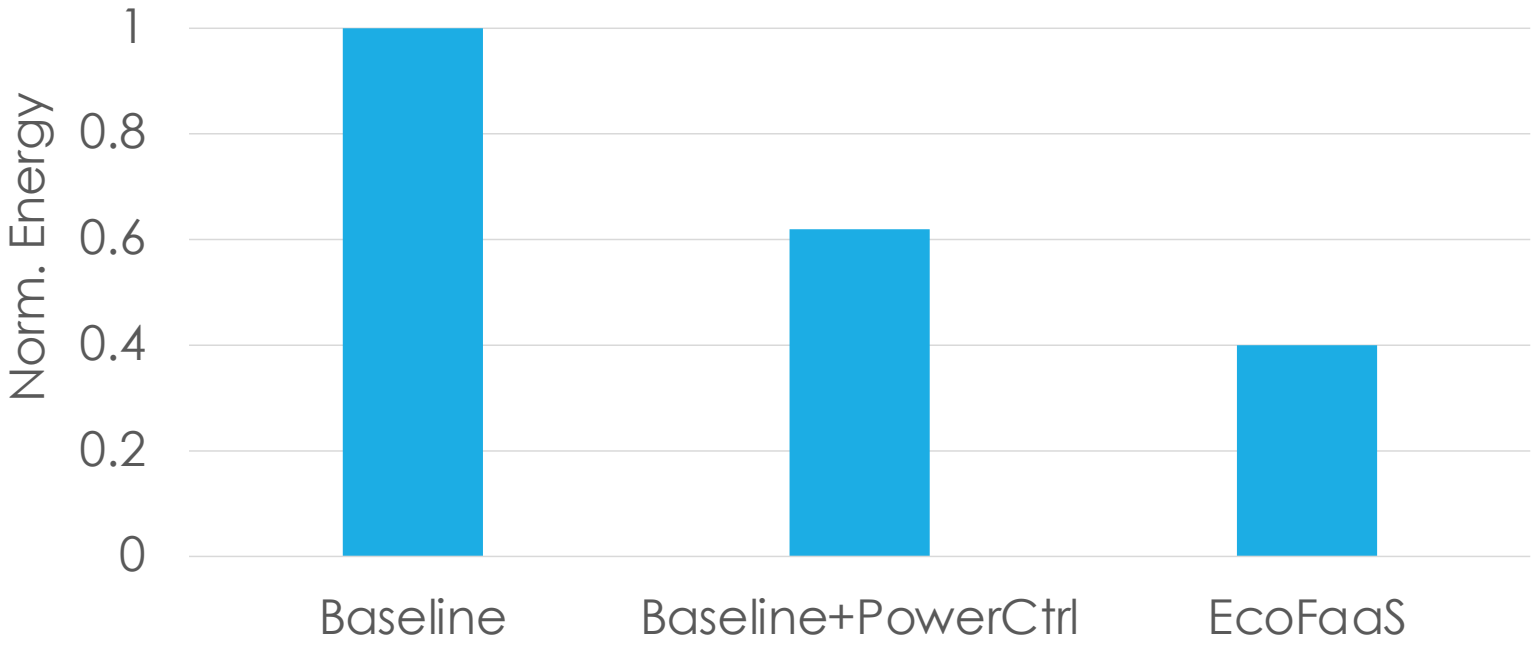
Re-configuring the Pools with EcoFaaS



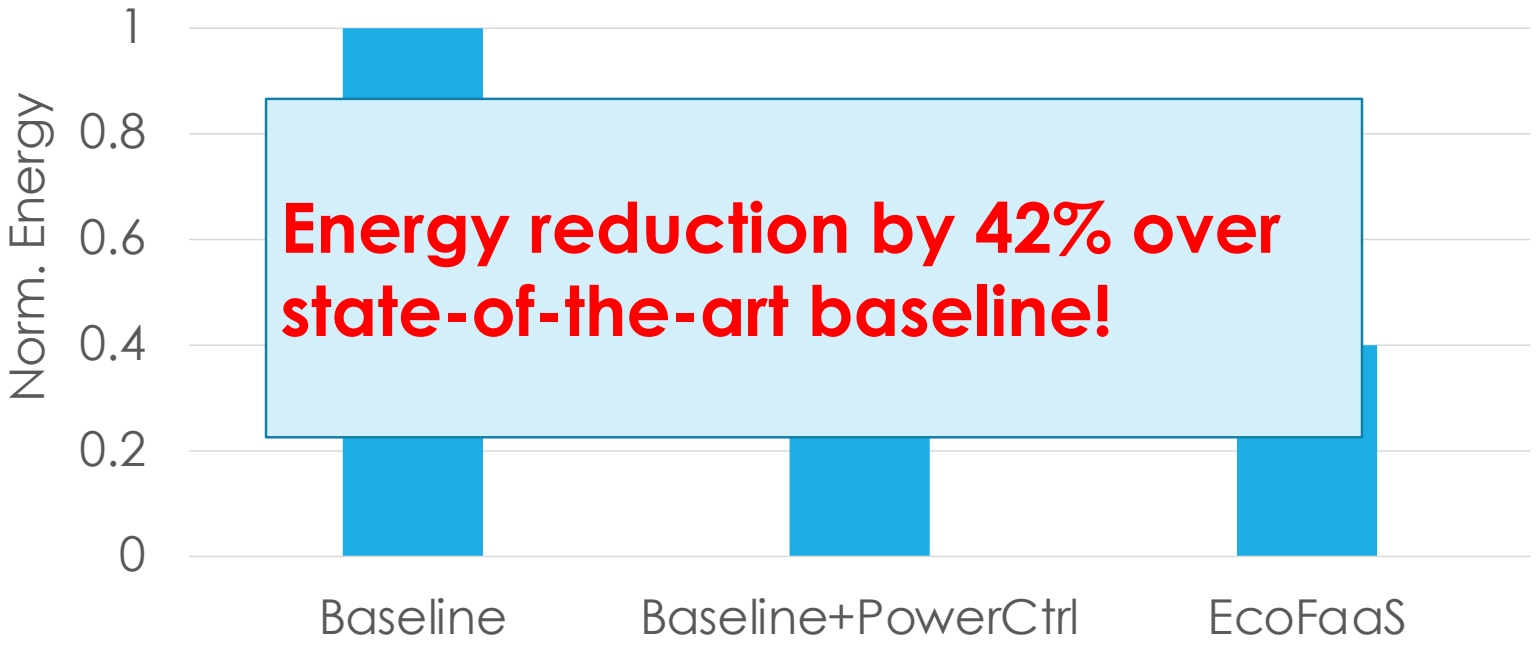
Evaluation Methodology

- Cluster with 5 Intel Xeon servers (20 cores, each)
- Platforms: OpenWhisk and KNative
- Systems evaluated
  - **Baseline:** *state-of-the-art* serverless platform (MXFaaS ISCA'23)
  - **Baseline+PowerCtrl:** Baseline + *state-of-the-art* power management (Gemini MICRO'20)
  - **EcoFaaS:** our proposal

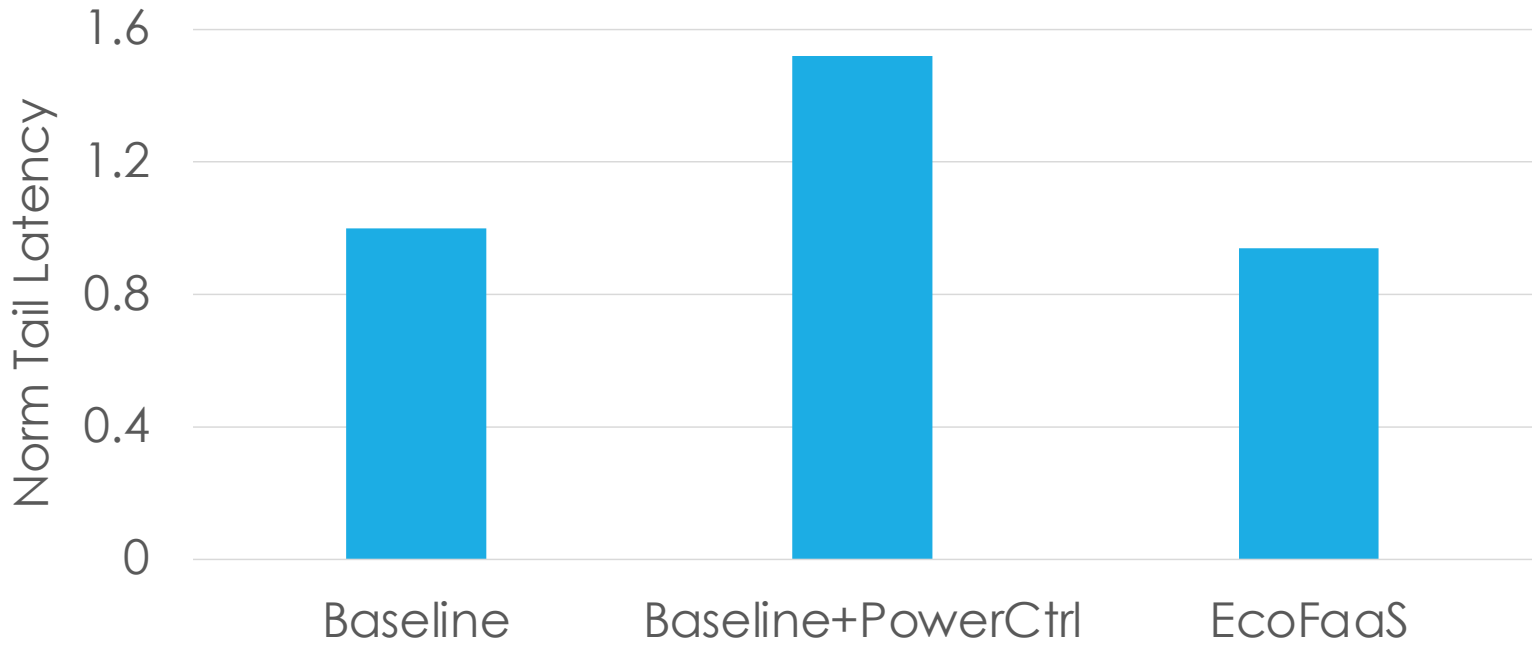
EcoFaaS Reduces Energy Consumption



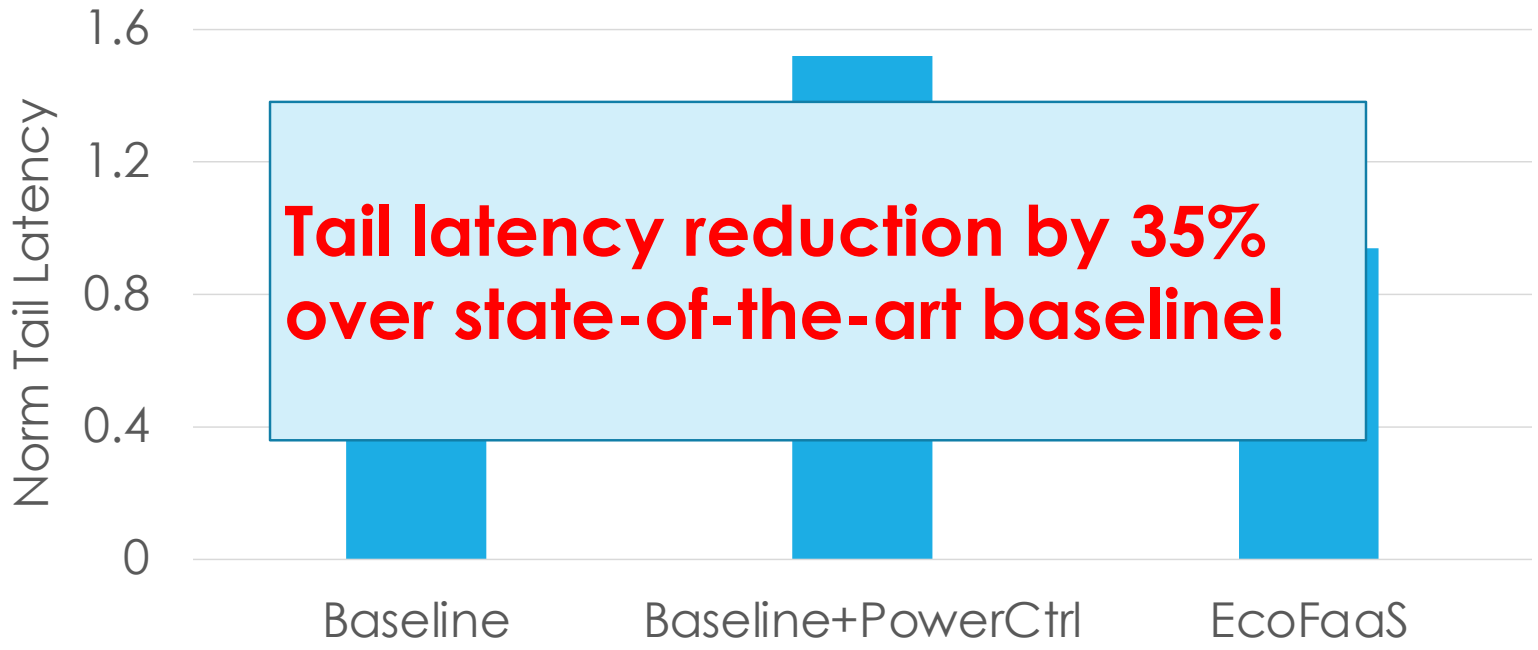
EcoFaaS Reduces Energy Consumption



EcoFaaS Reduces Tail Latency



EcoFaaS Reduces Tail Latency



Conclusion

- Serverless computing beneficial, but current execution energy-inefficient
- Propose **EcoFaaS** – SLO-driven energy-efficient serverless system
- Reduces energy consumption by 42%, and tail-latency by 35%

Questions?

EcoFaaS: Rethinking the Design of Serverless Environments for Energy Efficiency

ISCA 2024

Jovan Stojkovic, Nikoleta Iliakopoulou, Tianyin Xu, Hubertus Franke*, Josep Torrellas

University of Illinois at Urbana-Champaign

*IBM Research