FIRST: Exploiting the Multi-Dimensional Attributes of Functions for Power-Aware Serverless Computing

*Lu Zhang, Chao Li, **Xinkai Wang**, Weiqi Feng, Zheng Yu, Quan Chen, Jingwen Leng, Minyi Guo, Pu Yang, Shang Yue*

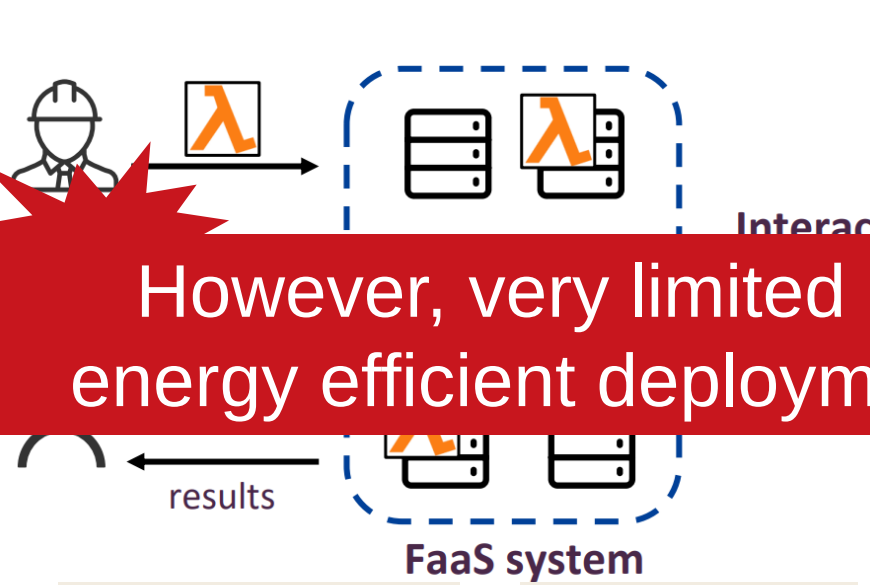
2023.5
Florida, USA

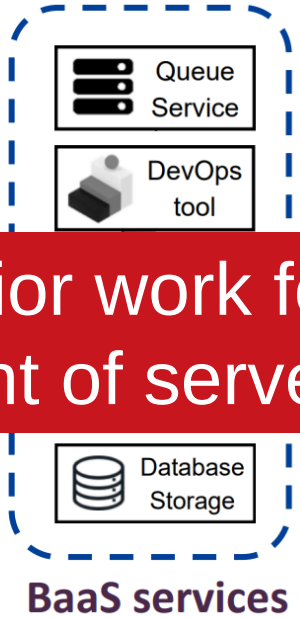
上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Contents

Contents

Background: Serverless Computing





Fine-grained Resource Management

FaasFlow, ATC'22
MLP, IPDPS'22

Function Collocation

Servermore, SoCC'21
Stretch, HPCA'19

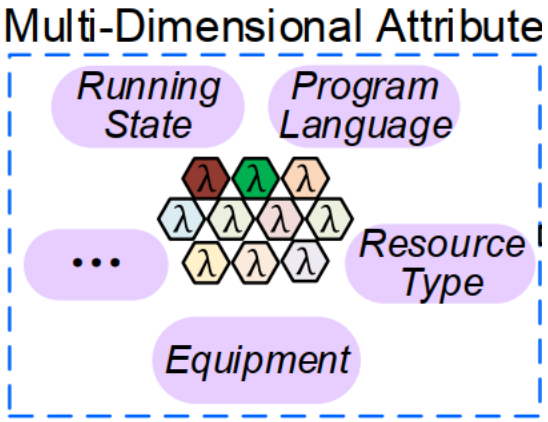
Pay-as-you-go

Auto-scaling

However, very limited prior work focuses on the energy efficient deployment of serverless functions.

Very limited work focuses on the energy efficient deployment of serverless functions considering the multi-dimensional performance-power behaviors.

Key Implications of Multi-dimensional Attributes

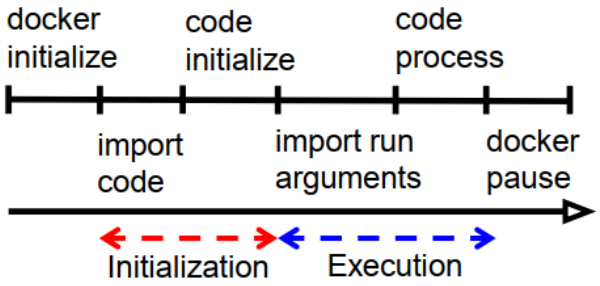


Serverless functions can be described from three levels of attributes.

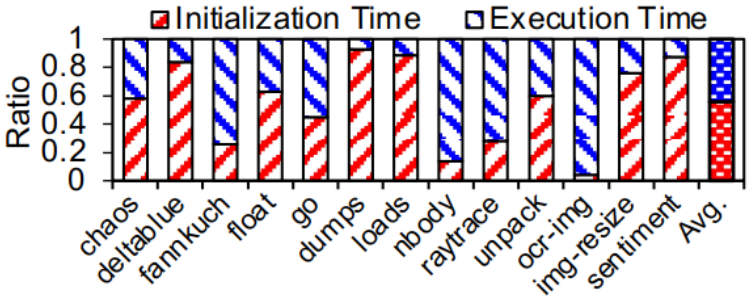
- Software Attributes
- Middleware Attributes
- Hardware Attributes

All dimensions of attributes with best performance-per-watt constructs optimal operating point (OOP) of functions.

Software Attribute: Function Phase Matters



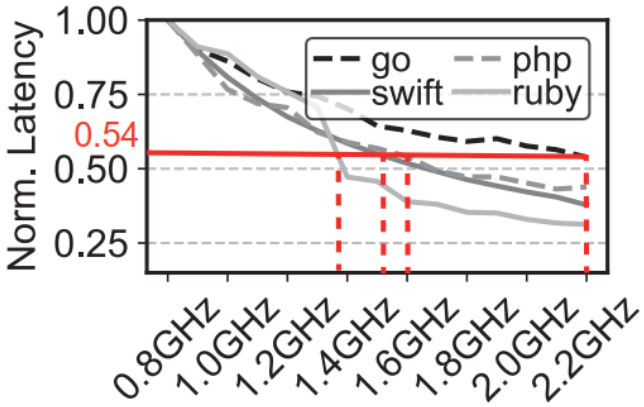
(a) The phases of functions



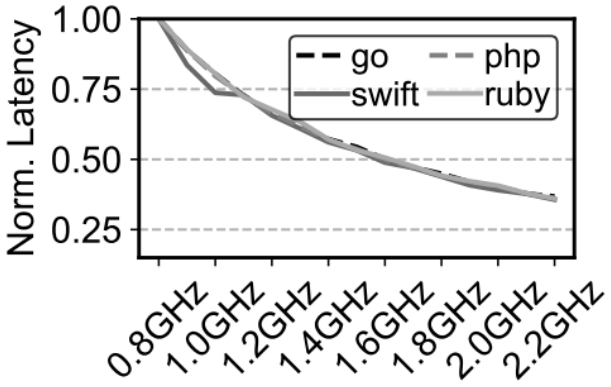
(b) Duration breakdown of functions

One needs to treat the initialization and execution phases differently when managing the energy of serverless functions.

Middleware Attribute: Language Runtime Matters



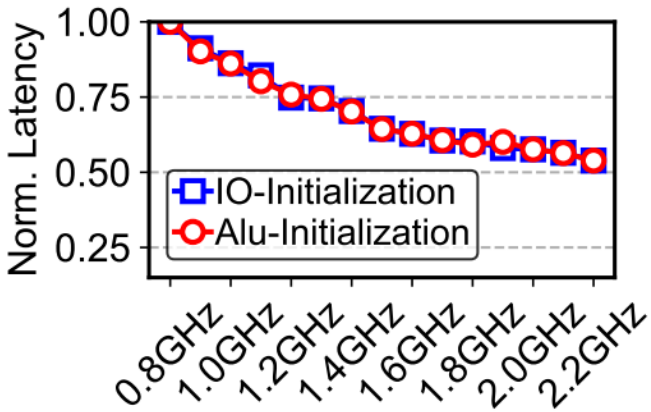
(a) Initialization phase



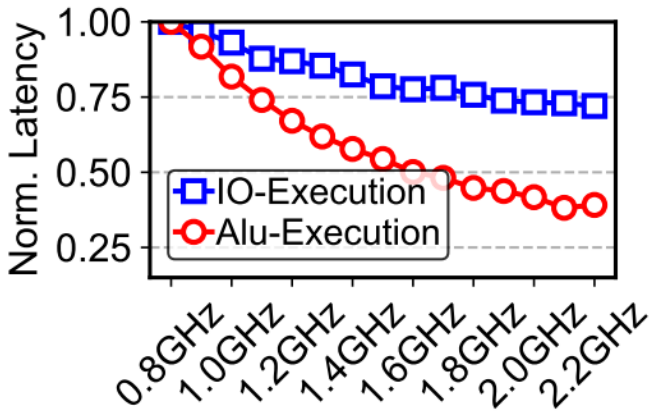
(b) Execution phase

A function's performance-power behaviors are language specific in the initialization phase while influenced less by language type in the execution phase.

Hardware Attribute: Resource Type Matters



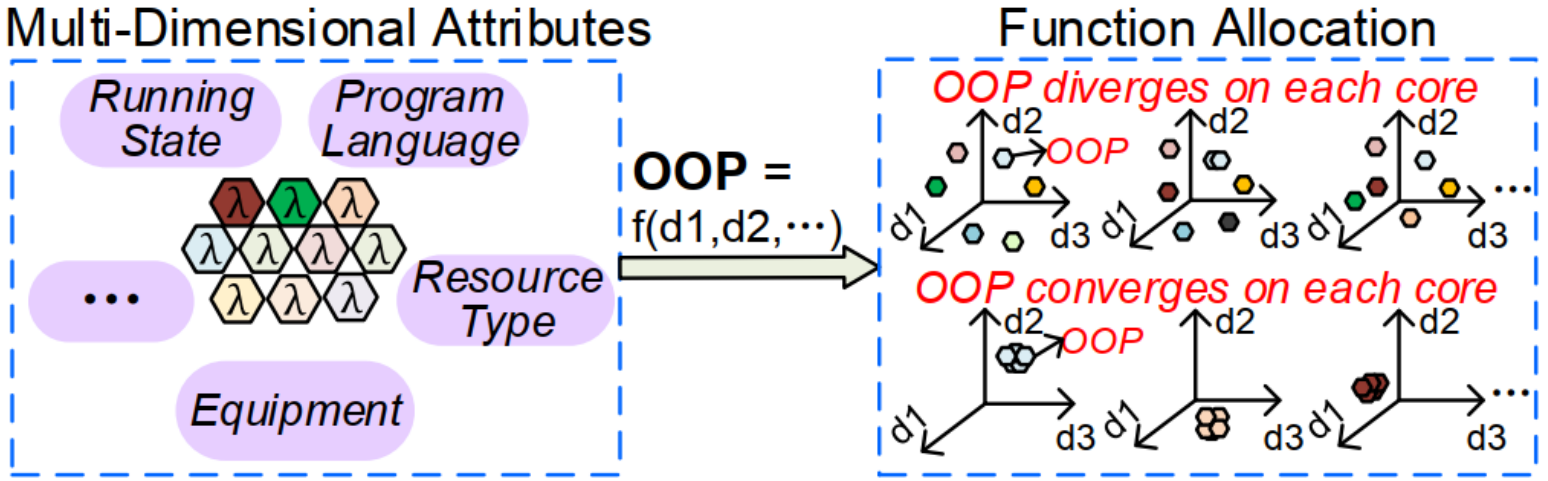
(a) Initialization phase



(b) Execution phase

Resource type has negligible influence during function initialization, but it dominates when a function is executing.

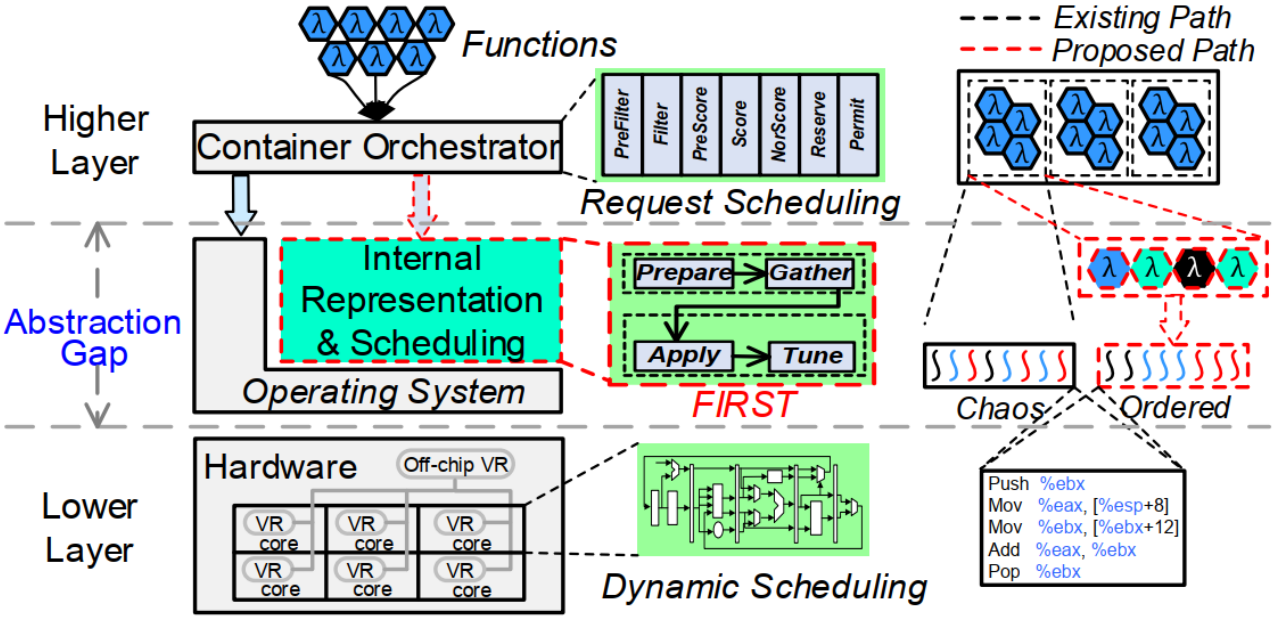
Challenge: Optimal Operating Point Divergence



- **OOP Divergence**: Multiple functions with varied OOPs collocate on a single processing core.
- Can't provide μs-scale power adjustment for each function.
- If OOP converges on each core, we can set the optimal power level for each group of functions to maximize energy efficiency!

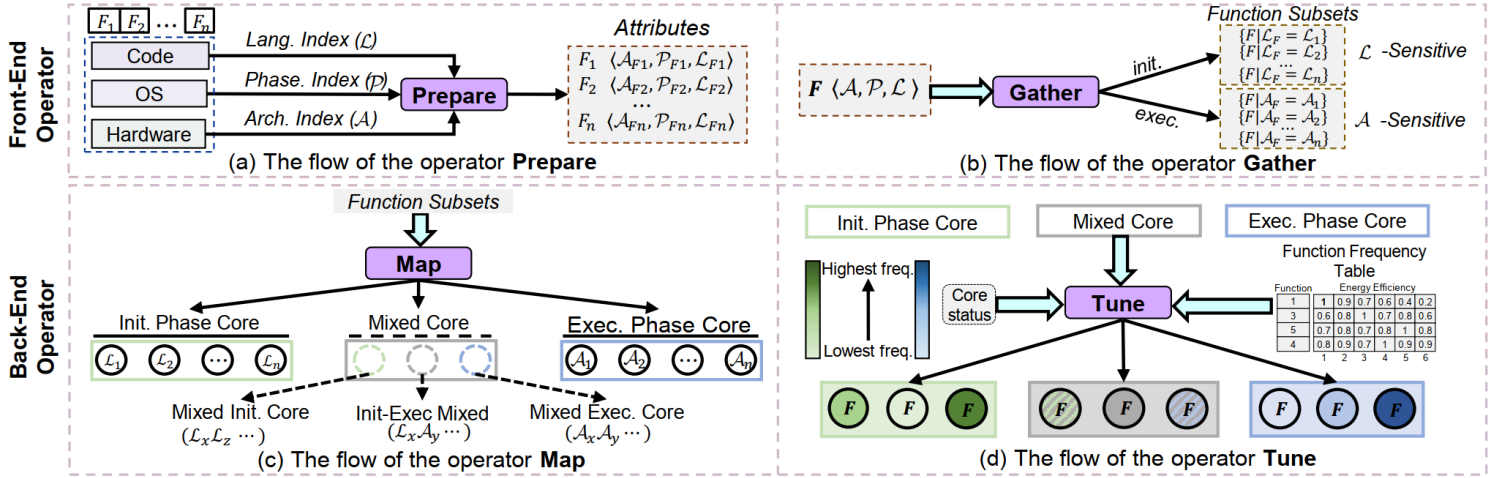
Contents

How to achieve OOP convergence



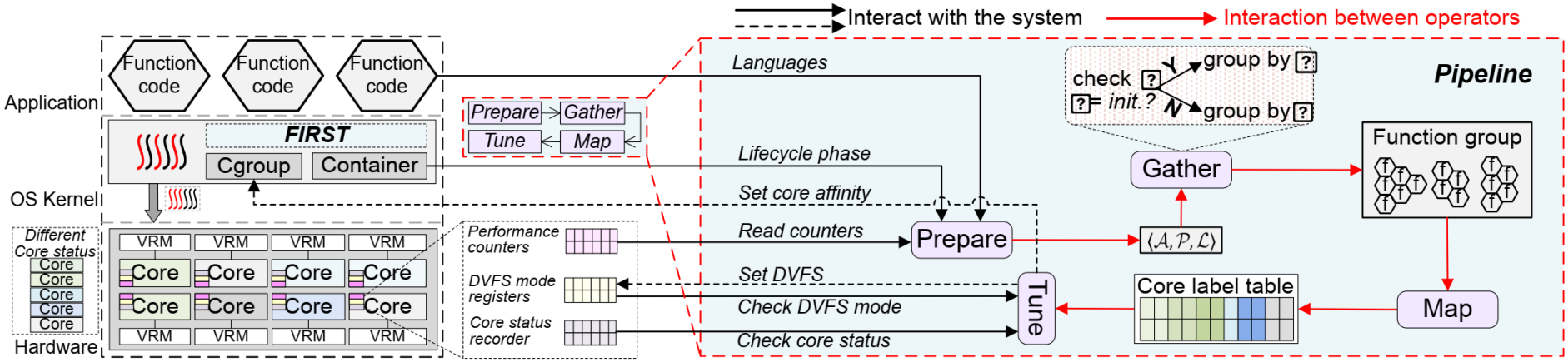
- Function Internal Representation(FIR): a new abstraction layer
- IR-based meta-scheduling (IRS): fine-tune the orchestration process
- OOP convergence: make core serve functions with converged OOP

IRS Design: Pipeline-like Workflow



- Front-end operators are responsible for information gathering and analysis.
  - *Prepare*: extracting the key information of functions
  - *Gather*: identifying the best resource sharing schemes
- Back-end operators are responsible for controlling functions' execution.
  - *Map*: mapping function subsets to cores
  - *Tune*: determining appropriate power levels for each CPU

Implementation and Optimization



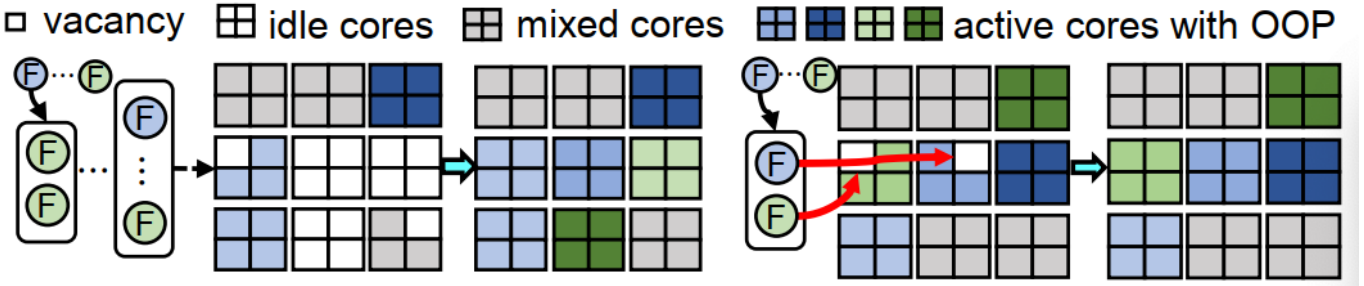
The full system architecture of applying FIRST.

- Read counters and runtimes to get internal representation.
- Associate function group to cores with core label table.
- Use dynamic voltage and frequency scaling to quickly adjust core state.

It is often necessary to make subtle yet non-trivial enhancements to the meta-scheduling pipeline.

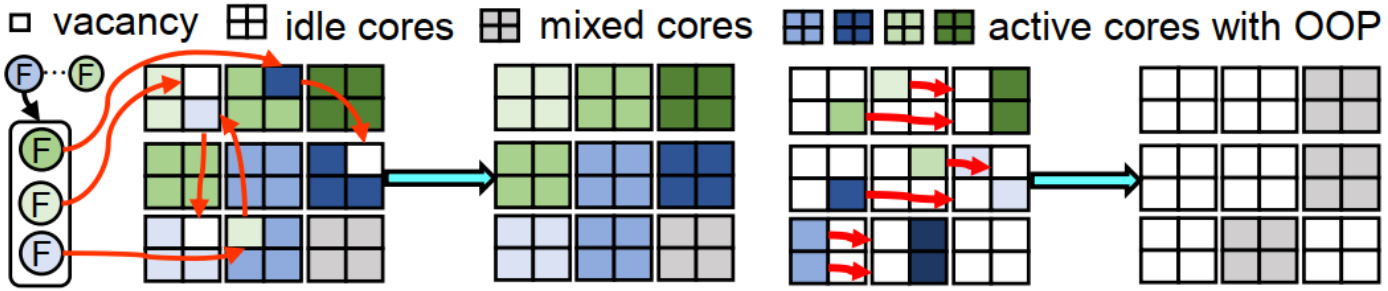
Realistic Operator Enhancement

Front-End Enhancement



- Adjust the pipeline workflow to tune the frontend execution.

Back-End Enhancement

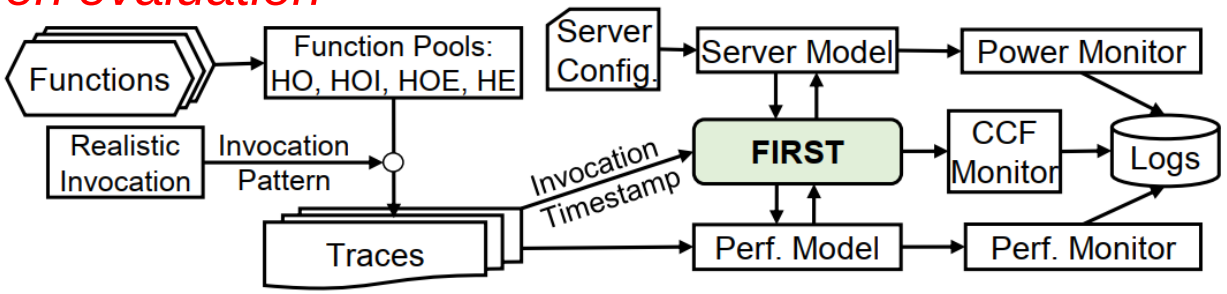


- Trade-off between OOP convergence and power saving.

Contents

Experimental Methodologies

Trace-driven evaluation



Characterization functions

Function	Description	Runtime
markdown	Renders the markdown text to HTML	python
img-resize	Resize images to icons	nodejs
sentiment	Sentiment analysis of text	python
ocr-img	Find text in images using OCR	nodejs
autocomplete	Autocomplete the string from a corpus	nodejs
FileIO	IO-intensive function	go
ALU	CPU-intensive function	go/ruby/swift/php

Evaluated function pools

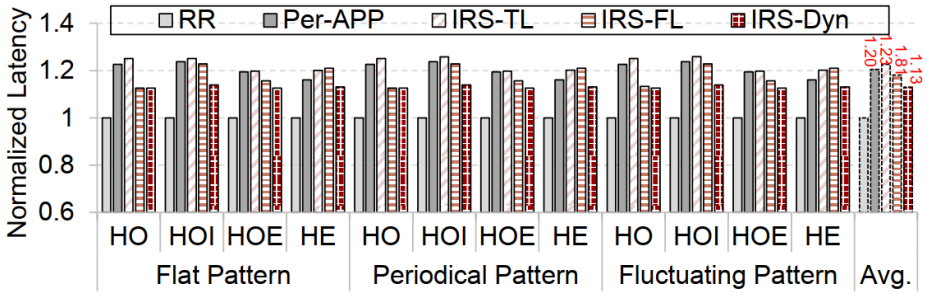
Function Pool	Abbr.	Functions
Homogeneous Pool ($FCF = 0$)	HO	Functions with $var(OOP)_{i\&e} = 0$
Less Homogeneous Pool ($0 < FCF < 1$)	HOI	Functions with $var(OOP)_i = 0, var(OOP)_e > 0$
	HOE	Functions with $var(OOP)_i > 0, var(OOP)_e = 0$
Heterogeneous Pool ($FCF = 1$)	HE	All functions combined $var(OOP)_i > 0, var(OOP)_e > 0$

Evaluated schemes

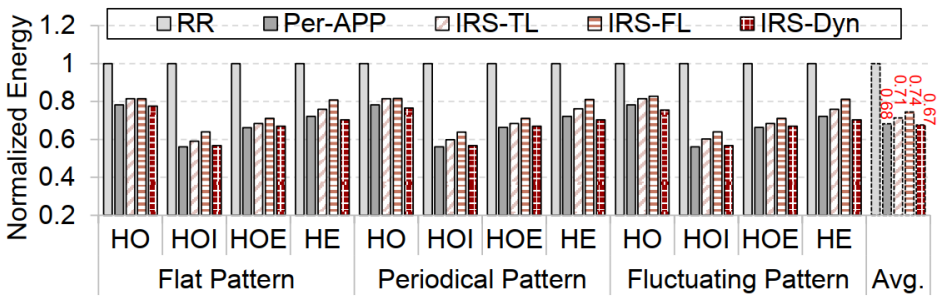
Mechanism	Description
PerfFst	Performance-first scheduling scheme for ideal performance
Per-APP	Fine-grained control considering function as a whole application
IRS-TL	FIRST in Tidal-Lane mode (enhanced `Prepare+`)
IRS-FL	FIRST in Fast-Lane mode (enhanced `Gather+`)
IRS-Dyn	FIRST with enhanced front-end operators and `Map+`

Result: Effectiveness of FIRST

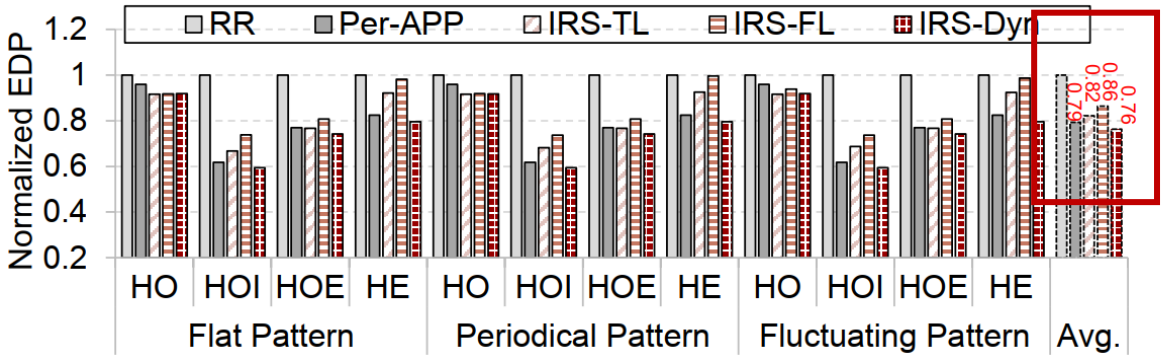
Comparison of the latency



Comparison of the energy



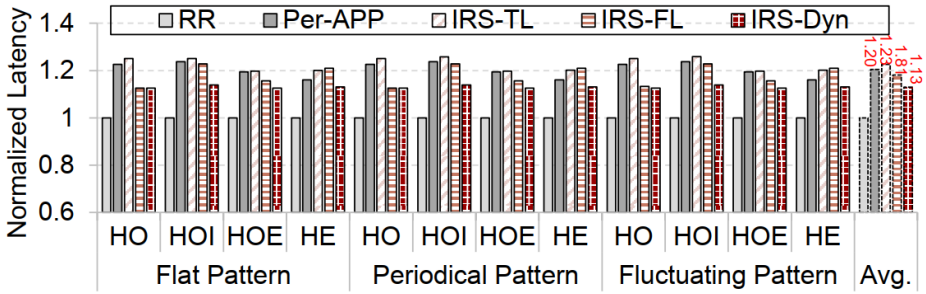
Comparison of the EDP



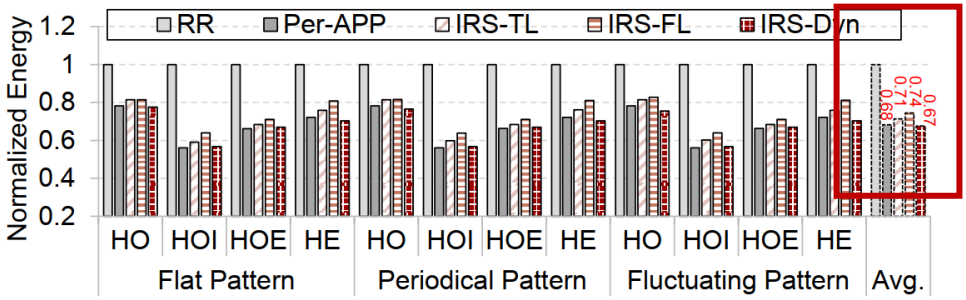
Our design can achieve better energy efficiency with little performance loss.

Result: Effectiveness of FIRST

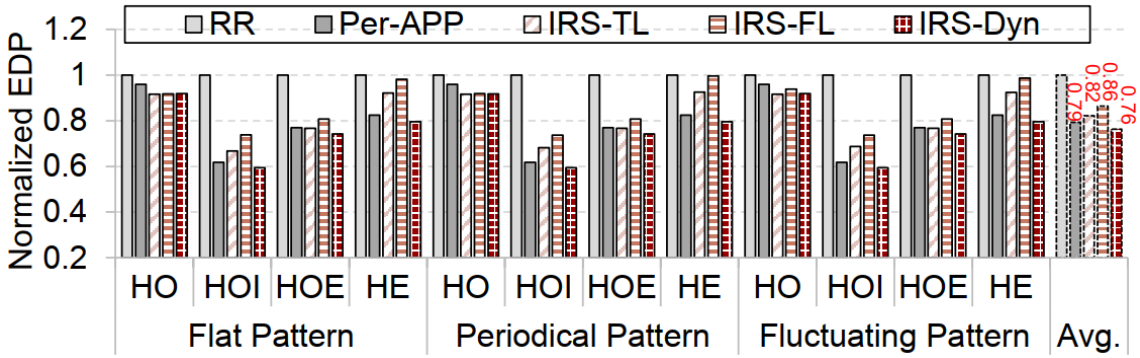
Comparison of the latency



Comparison of the energy



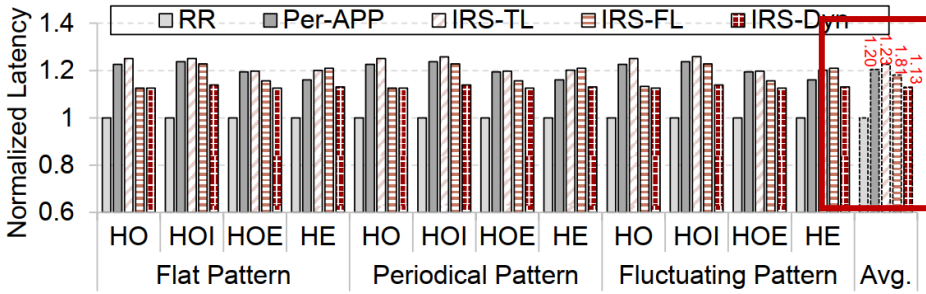
Comparison of the EDP



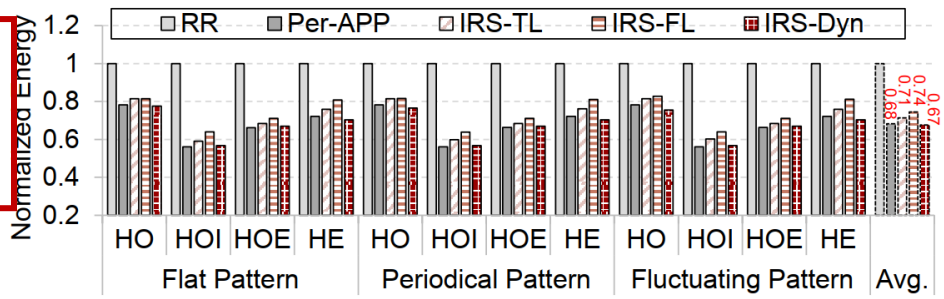
Our design can achieve better energy efficiency with little performance loss.

Result: Effectiveness of FIRST

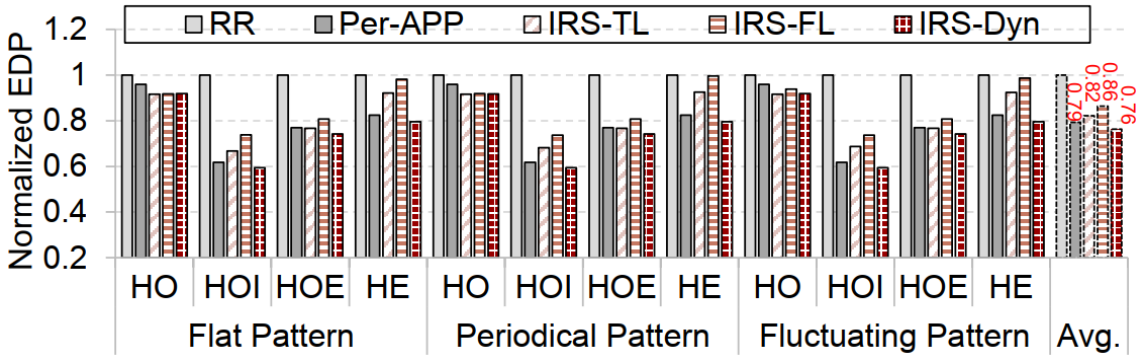
Comparison of the latency



Comparison of the energy



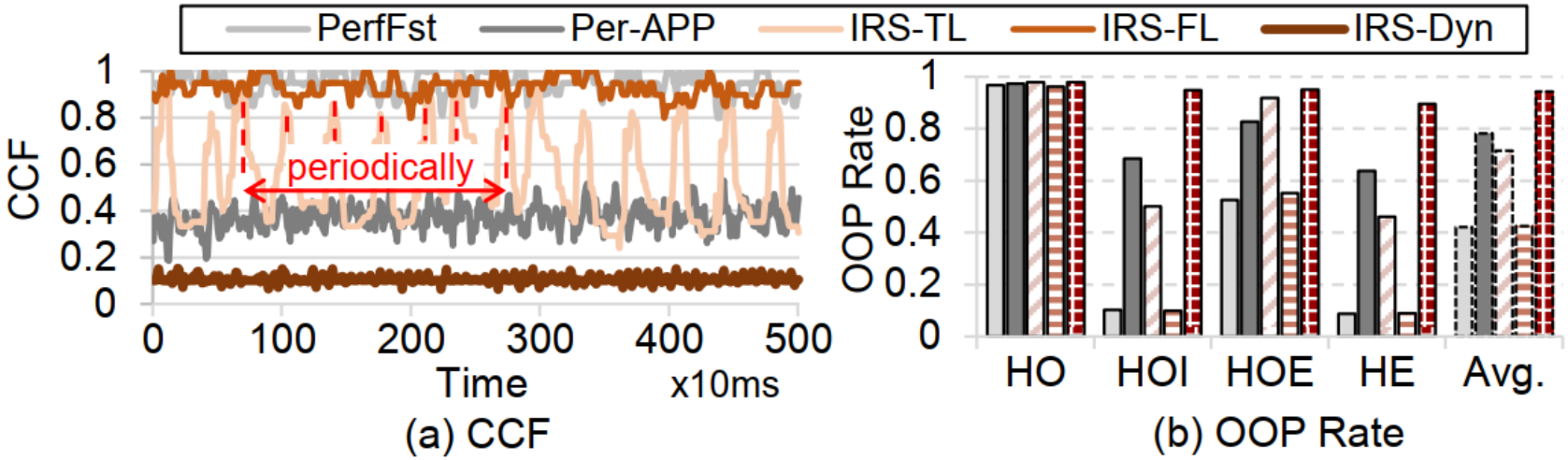
Comparison of the EDP



Our design can achieve better energy efficiency with little performance loss.

Result: OOP Convergence

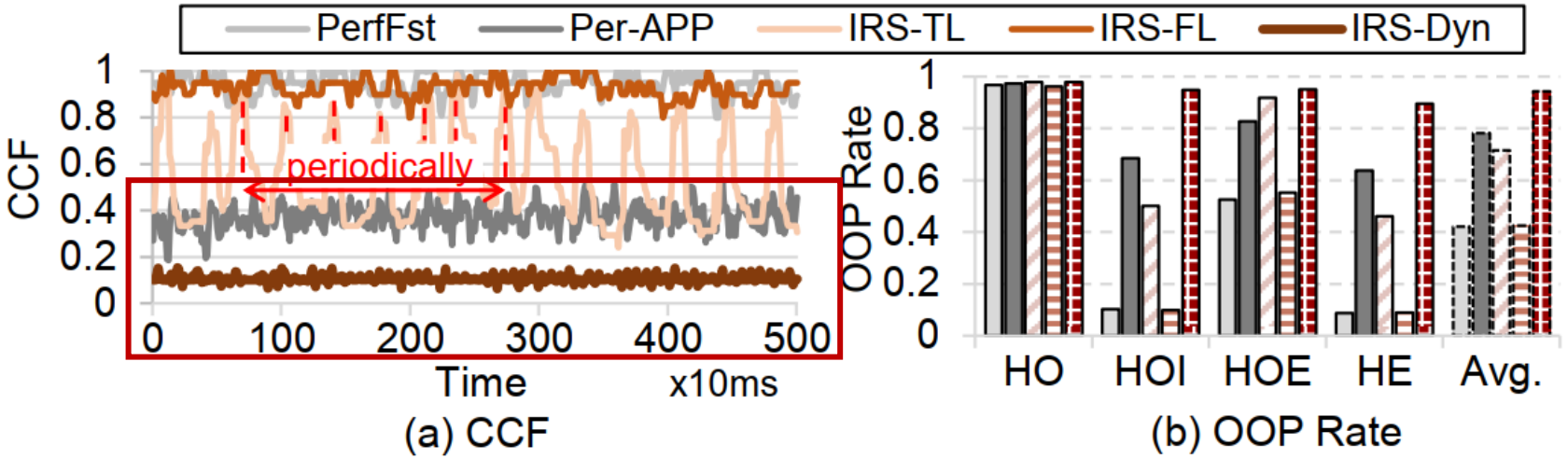
Core Chaos Factor and Optimal Operating Point Comparison



Our design achieves such improvements through improving core convergence and OOP convergence.

Result: OOP Convergence

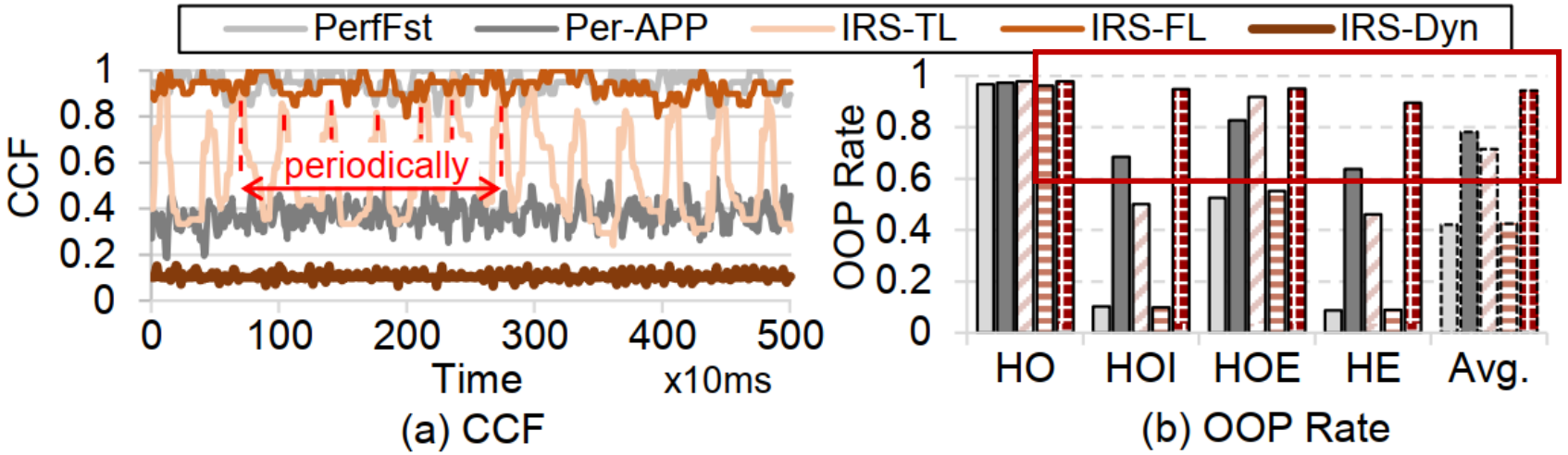
Core Chaos Factor and Optimal Operating Point Comparison



Our design achieves such improvements through improving core convergence and OOP convergence.

Result: OOP Convergence

Core Chaos Factor and Optimal Operating Point Comparison



Our design achieves such improvements through improving core convergence and OOP convergence.

Contents

Conclusion

- We analyze the multi-dimensional performance-power implications of serverless functions and demonstrate the rationale for maintaining a converged optimal operating point.
- We introduce FIRST, a novel mechanism for fine-tuning the function placement process with a pipeline-like workflow and enhance FIRST to support more flexible function power management.
- We build a proof-of-concept testbench of FIRST and show that it improves energy efficiency by more than 24% with minor performance overhead.
- Energy efficiency of serverless platform is promising and we plan to explore architectures to support fine-grained power management designs.

Thank You!

FIRST: Exploiting the Multi-Dimensional Attributes of Functions for Power-Aware Serverless Computing

*Lu Zhang, Chao Li, **Xinkai Wang**, Weiqi Feng, Zheng Yu, Quan Chen, Jingwen Leng, Minyi Guo, Pu Yang, Shang Yue*
unbreakablewxk@sjtu.edu.cn