EcoLife: Carbon-Aware Serverless Function Scheduling for Sustainable Computing


Yankai Jiang


Rohan Basu Roy


Baolin Li


Devesh Tiwari

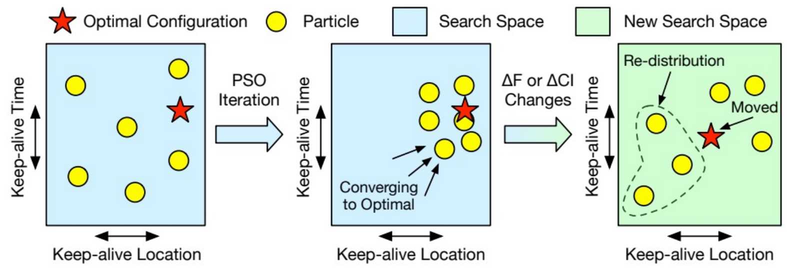
Northeastern University

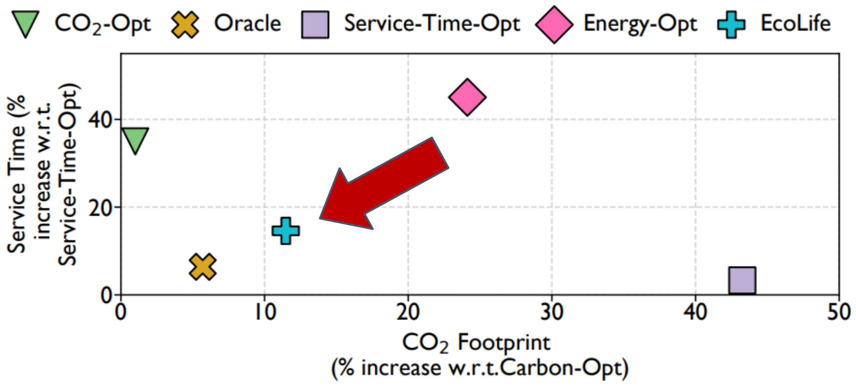
EcoLife: Executive Summary

- ✓ **Introducing a new problem space:** Carbon footprint of serverless computing model.
- ✓ **Key Idea of EcoLife:** Exploit a mix of old and new hardware to reduce the carbon footprint but also, achieve high performance.
- ✓ **Novel Particle Swarm Optimization (PSO) based optimizer** to achieve near-Oracle results.

EcoLife

Particle Swarm Optimization





Serverless Computing Model Becoming Increasingly Popular for HPC Workflows

funcX: A Federated Function Serving Fabric for Science

Ryan Chard, Argonne National Laboratory
Yadu Babuji, University of Chicago
Zhuozhao Li, University of Chicago
Tyler Skluzacek, University of Chicago
Anna Woodard, University of Chicago
Ben Blaiszik, University of Chicago
Ian Foster, Argonne National Laboratory and University of Chicago
Kyle Chard, University of Chicago and Argonne National Laboratory

HPDC '20

DayDream: Executing Dynamic Scientific Workflows on Serverless Platforms with Hot Starts

Rohan Basu Roy, *Northeastern University*, Boston, MA, USA, rohanbasuroy@gmail.com
Tirthak Patel, *Northeastern University*, Boston, MA, USA, patel.ti@northeastern.edu
Devesh Tiwari, *Northeastern University*, Boston, MA, USA, devesh.dtiwari@gmail.com

SC '22

WUKONG: A Scalable and Locality-Enhanced Framework for Serverless Parallel Computing

Benjamin Carver, George Mason University, bcarver2@gmu.edu
Jingyuan Zhang, George Mason University, jzhang33@gmu.edu
Ao Wang, George Mason University, awang24@gmu.edu
Ali Anwar, IBM Research–Almaden, Ali.Anwar2@ibm.com
Panruo Wu, University of Houston, pwu7@uh.edu
Yue Cheng, George Mason University, yuecheng@gmu.edu

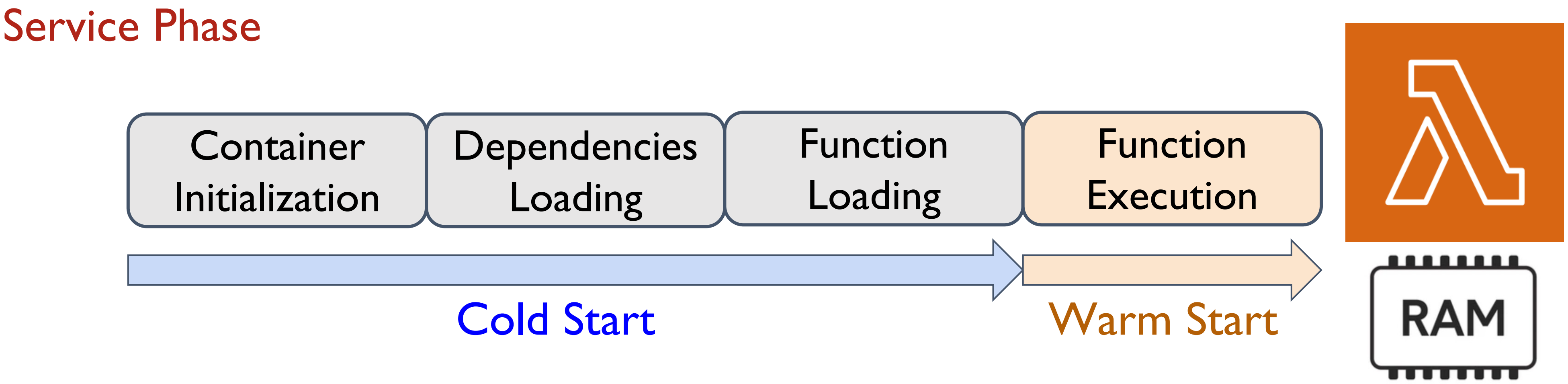
SoCC '20

Mashup: Making Serverless Computing Useful for HPC Workflows via Hybrid Execution

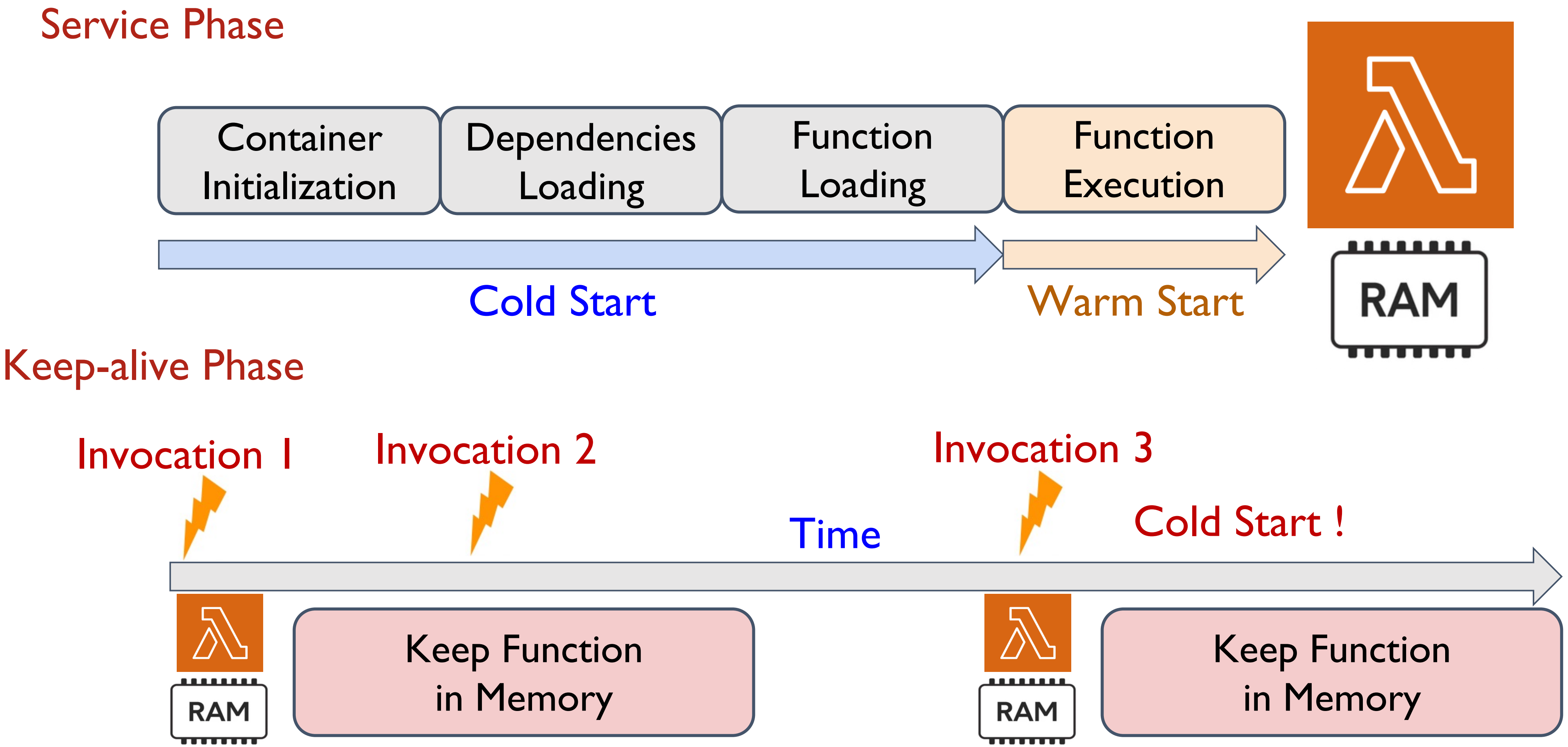
Rohan Basu Roy, Northeastern University
Tirthak Patel, Northeastern University
Vijay Gadepally, MIT Lincoln Laboratory
Devesh Tiwari, Northeastern University

PPoPP '22

Serverless Providers Keep Functions in Memory to Avoid Cold Start Overhead



Serverless Providers Keep Functions in Memory to Avoid Cold Start Overhead



Computing Carbon Footprint
An Emerging Challenge


The Washington Post
Democracy Dies in Darkness

CLIMATE Environment Weather Climate Solutions Climate Lab Green Living Business of Climate

World is on brink of catastrophic warming, U.N. climate change report says

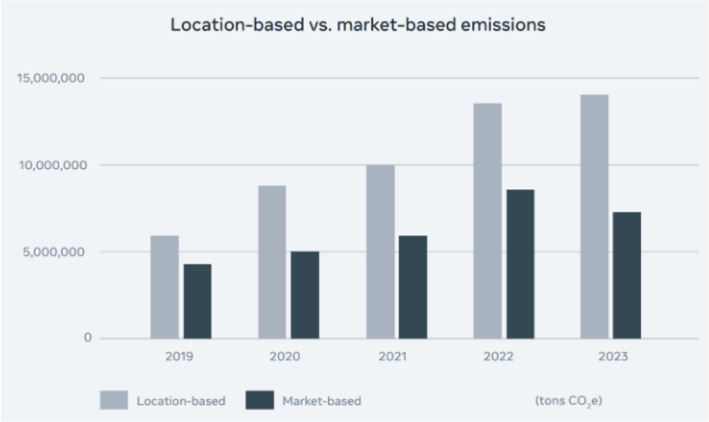
A dangerous climate threshold is near, but 'it does not mean we are doomed' if swift action is taken, scientists say

The Carbon Footprint of Amazon, Google, and Facebook Is Growing

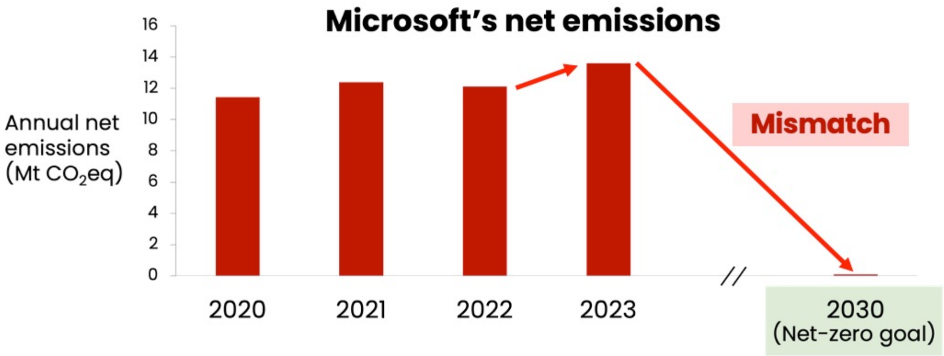
How cloud computing—and especially AI—threaten to make climate change worse

Microsoft builds first datacenters with wood to slash carbon emissions

Meta Carbon Spotlight '24



Microsoft Sustainability Report '24



HPC and Datacenter Carbon Footprint Efforts

Chasing Carbon (HPCA '21)
ACT (ISCA '22)
Carbon Explorer (ASPLOS '22)

$$\text{Operational } CO_2 = \text{Energy} \times \text{CI}$$

$$\text{Embodied } CO_2 = \frac{\text{Time}}{\text{LifeTime}} \times \text{Embodied}_{\text{Hardware}}$$

HPC and Datacenter Carbon Footprint Efforts

Chasing Carbon (HPCA '21)
ACT (ISCA '22)
Carbon Explorer (ASPLOS '22)

$$\text{Operational CO}_2 = \text{Energy} \times \text{CI}$$

$$\text{Embodied CO}_2 = \frac{\text{Time}}{\text{LifeTime}} \times \text{Embodied}_{\text{Hardware}}$$

CLOVER: Toward Sustainable AI with Carbon-Aware Machine Learning Inference Service

Baolin Li, Northeastern University
Siddharth Samsi, MIT Lincoln Laboratory
Vijay Gadepally, MIT Lincoln Laboratory
Devesh Tiwari, Northeastern University

SC '23, ML Inference

Toward Sustainable HPC: Carbon Footprint Estimation and Environmental Implications of HPC Systems

Baolin Li, Northeastern University
Rohan Basu Roy, Northeastern University
Daniel Wang, Northeastern University
Siddharth Samsi, MIT
Vijay Gadepally, MIT
Devesh Tiwari, Northeastern University

SC '23, HPC Carbon

FairyWREN: A Sustainable Cache for Emerging Write-Read-Erase Flash Interfaces

Sara McAllister, Yucong "Sherry" Wang, Benjamin Berg*, Daniel S. Berger[†]
George Amvrosiadis, Nathan Beckmann, Gregory R. Ganger
Carnegie Mellon University **UNC Chapel Hill* *[†]Microsoft Azure and University of Washington*

OSDI '24, Memory and Cache

HPC and Datacenter Carbon Footprint Efforts

Chasing Carbon (HPCA ’21)
ACT (ISCA ’22)
Carbon Explorer (ASPLOS ’22)

$$\text{Operational CO}_2 = \text{Energy} \times \text{CI}$$

$$\text{Embodied CO}_2 = \frac{\text{Time}}{\text{LifeTime}} \times \text{Embodied}_{\text{Hardware}}$$

CLOVER: Toward Sustainable AI with Carbon-Aware Machine Learning Inference Service

Baolin Li, Northeastern University
Siddharth Samsi, MIT Lincoln Laboratory
Vijay Gadepally, MIT Lincoln Laboratory
Devesh Tiwari, Northeastern University

SC ’23, ML Inference

Toward Sustainable HPC: Carbon Footprint Estimation and Environmental Implications of HPC Systems

Baolin Li, Northeastern University
Rohan Basu Roy, Northeastern University
Daniel Wang, Northeastern University
Siddharth Samsi, MIT
Vijay Gadepally, MIT
Devesh Tiwari, Northeastern University

SC ’23, HPC Carbon

FairyWREN: A Sustainable Cache for Emerging Write-Read-Erase Flash Interfaces

Sara McAllister, Yucong “Sherry” Wang, Benjamin Berg*, Daniel S. Berger†
George Amvrosiadis, Nathan Beckmann, Gregory R. Ganger

Carnegie Mellon University, **UNC Chapel Hill*, *†Microsoft Azure and University of Washington*

OSDI ’24, Memory and Cache

What about carbon footprint of serverless computing?

HPC and Datacenter Carbon Footprint Efforts

Chasing Carbon (HPCA '21)
ACT (ISCA '22)
Carbon Explorer (ASPLOS '22)

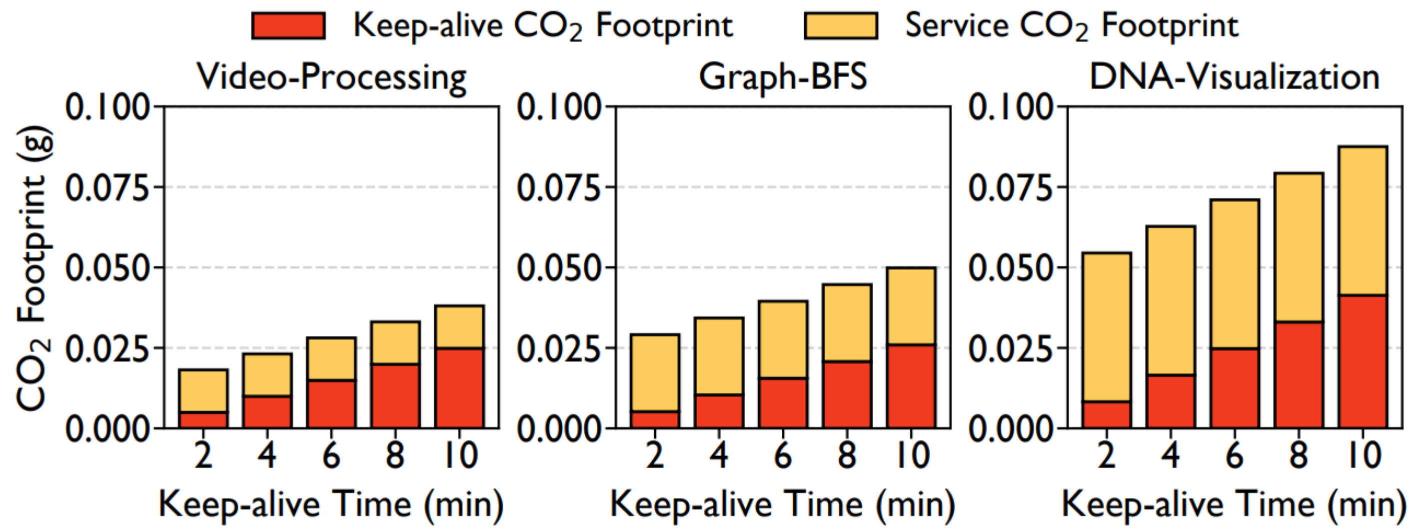
$$\text{Operational } CO_2 = \text{Energy} \times \text{CI}$$

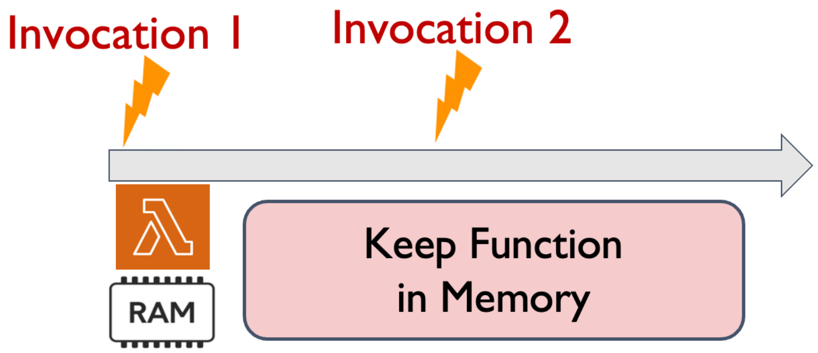
$$\text{Embodied } CO_2 = \frac{\text{Time}}{\text{LifeTime}} \times \text{Embodied}_{\text{Hardware}}$$

Carbon footprint is the dark secret of serverless computing. Currently, no carbon-aware solution for serverless workloads and systems!

What about carbon footprint of serverless computing?

Unique and Hidden Carbon Footprint of Serverless Computing





Serverless functions generate a significant carbon footprint during their keep-alive period — which is unique compared to the traditional non-serverless computing model. Longer keep-alive period leads to higher performance due to higher likelihood of warm starts.

Opportunity Embodied vs. Operational Carbon Trade-Off

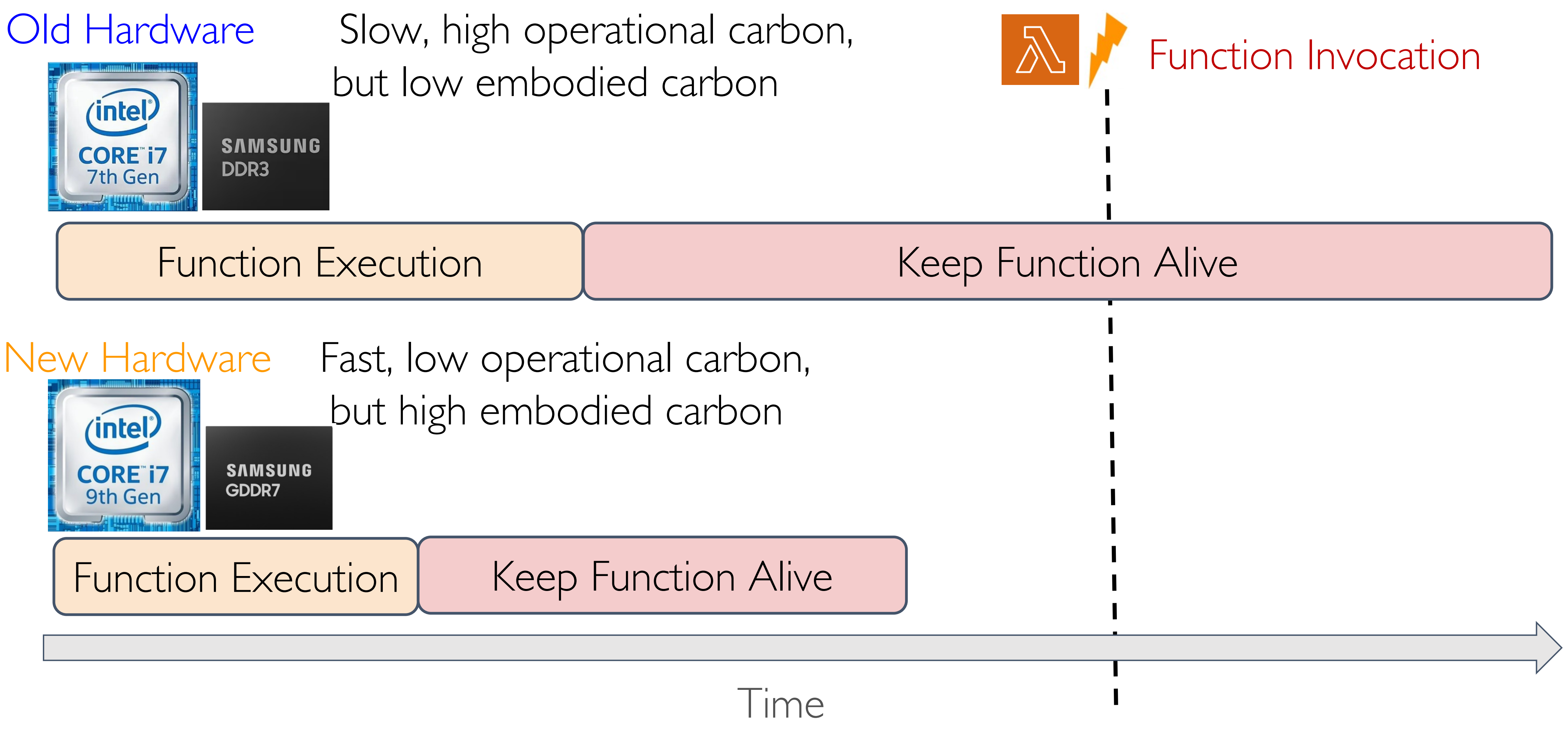
Old Hardware

Slow performance, high operational carbon, but low embodied carbon

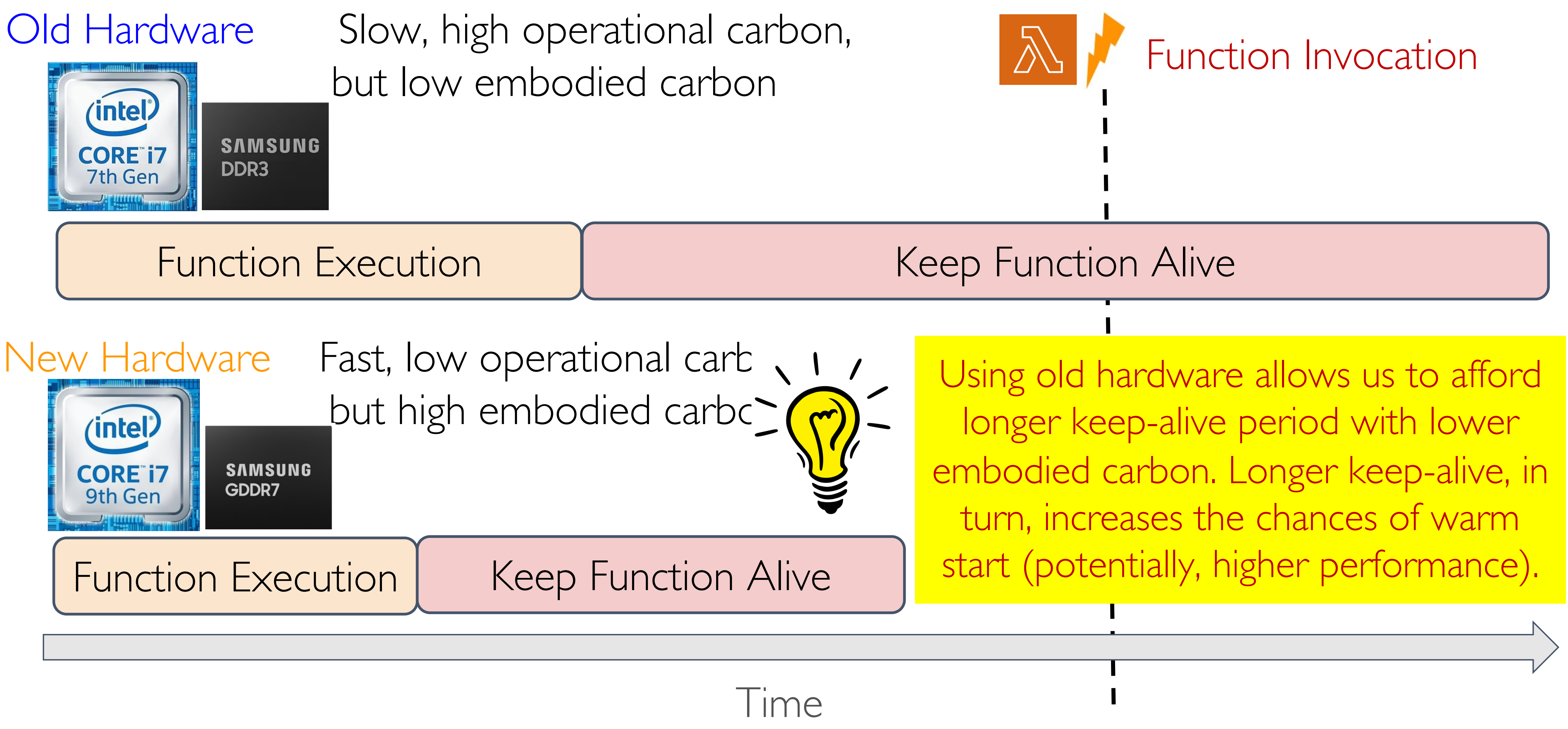
New Hardware

Fast performance, low operational carbon, but high embodied carbon

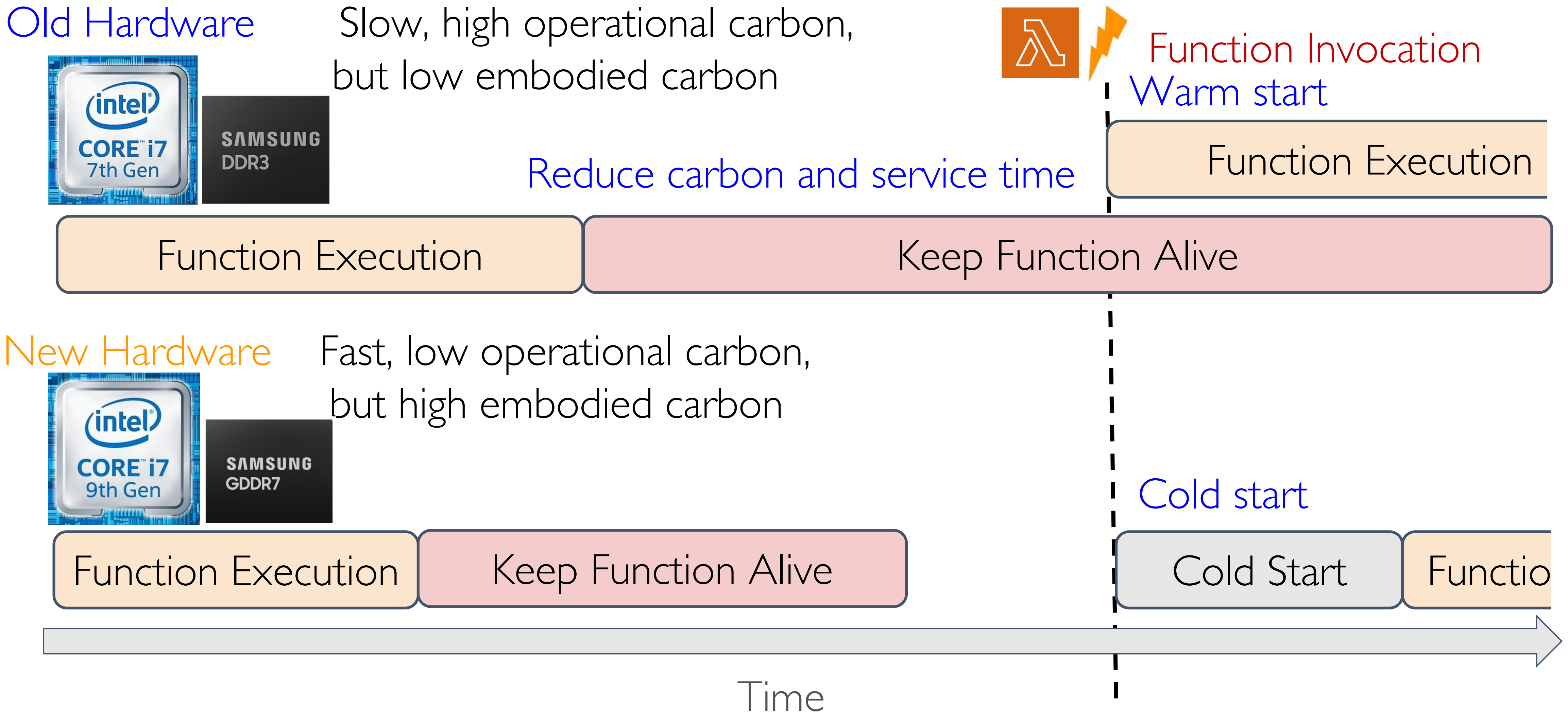
EcoLife's Old Hardware Use Enables New Opportunities



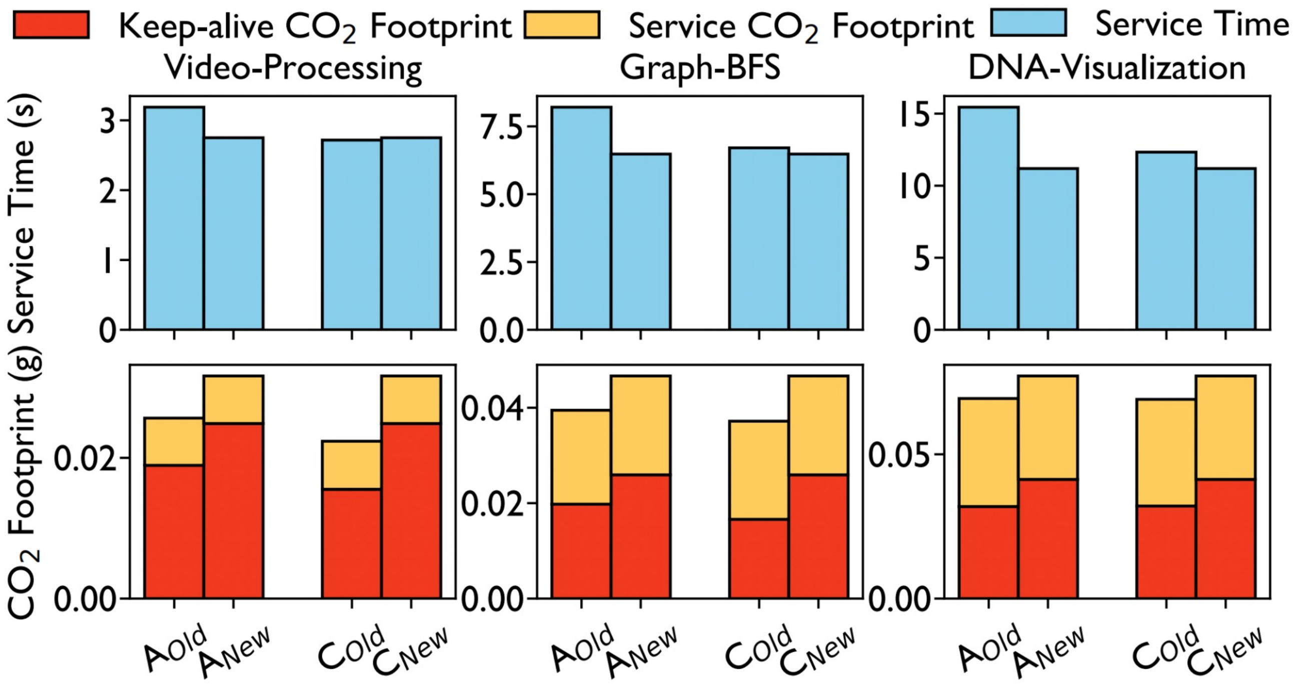
EcoLife's Old Hardware Use Enables New Opportunities



EcoLife's Old Hardware Use Enables New Opportunities



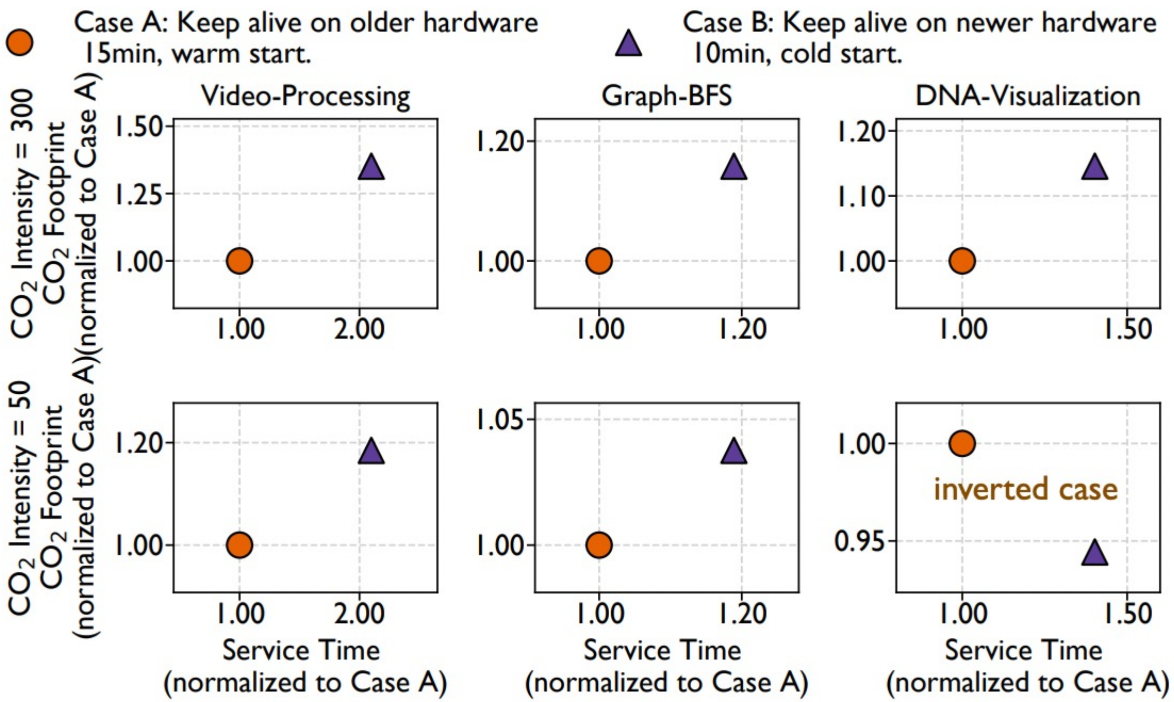
Carbon Opportunity of Multi-Generation Hardware Mix



Pair	Old/New	CPU Model (Year)	DRAM Model (Year)
$Pair_A$	A_{OLD}	Intel Xeon E5-2686 (2016)	Micron-512 (2018)
	A_{NEW}	Intel Xeon Platinum 8252C (2020)	Samsung-192 (2019)
$Pair_B$	B_{OLD}	Intel Xeon Platinum 8124M (2017)	Micron-192 (2018)
	B_{NEW}	Intel Xeon Platinum 8252C (2020)	Samsung-192 (2019)
$Pair_C$	C_{OLD}	Intel Xeon Platinum 8275L (2019)	Samsung-192 (2019)
	C_{NEW}	Intel Xeon Platinum 8252C (2020)	Samsung-192 (2019)

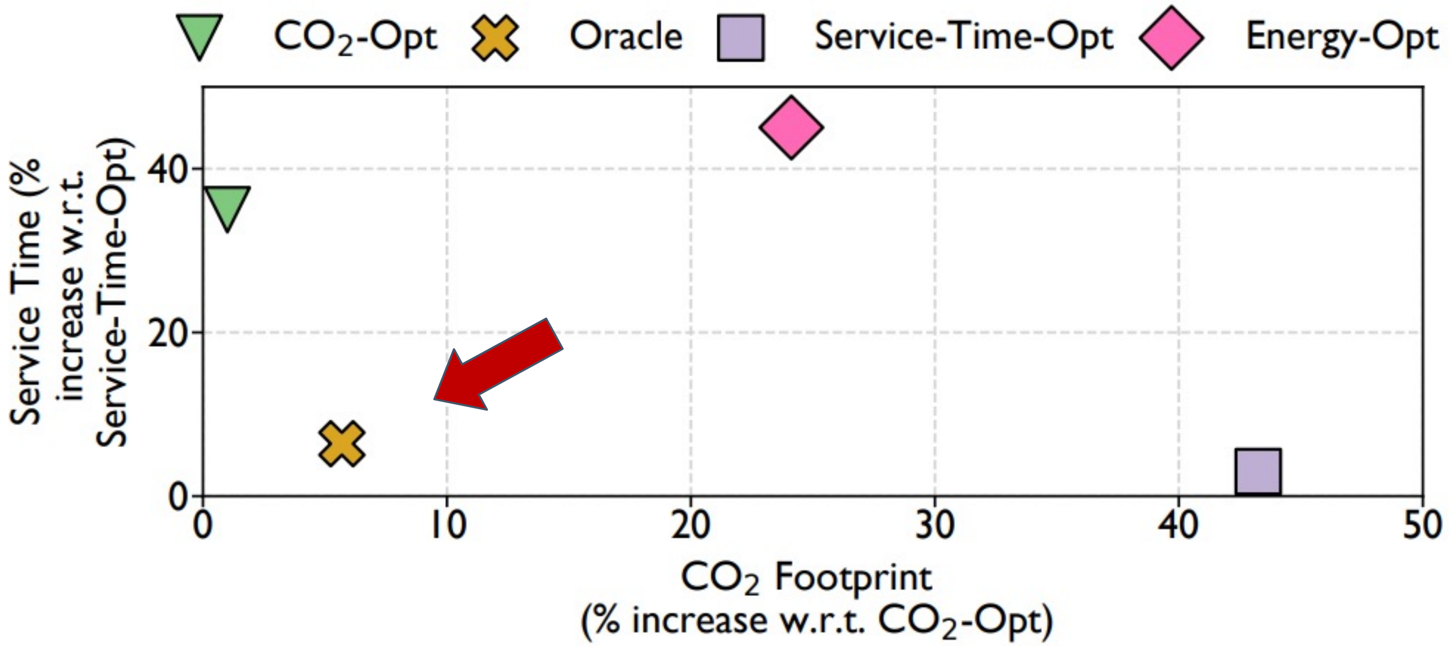
The use of relatively older-generation hardware, which inherently has a lower embodied carbon footprint, opens the opportunity to lower the carbon footprint while achieving high performance.

Exploiting Old and New Hardware Mix is Challenging



A longer keep-alive period on older-generation hardware can potentially reduce both service time and carbon footprint, but exploiting this opportunity depends on function characteristics and carbon intensity.

Exploiting Old and New Hardware Mix is Challenging



Co-optimization of service time and carbon footprint has significant potential, but is extremely challenging.

EcoLife Objectives and Key Ideas

EcoLife

objectives

Service Time

Carbon Footprint During Execution

$$\underset{l \in L,\ k \in \mathrm{KAT}}{\operatorname{argmin}} \ \lambda_s \frac{E[S_{f_{l,k}}]}{S_{f_{\max}}} + \lambda_c \frac{E[SC_{f_{l,k}}]}{SC_{f_{\max}}} + \lambda_c \frac{KC_{f_{l,k}}}{KC_{f_{k_{\max}}}}$$

Carbon Footprint During Keep-alive period

The goal of EcoLife is to determine the most suitable location (older-generation hardware or newer-generation hardware) and keep-alive periods for serverless functions.

How to Optimize for EcoLife Objectives?

EcoLife uses Particle Swarm Optimization (PSO) to determine the keep-alive time of functions.

Inertia Weight

Updated Velocity

Previous Velocity

Personal Best Position

Global Best Position

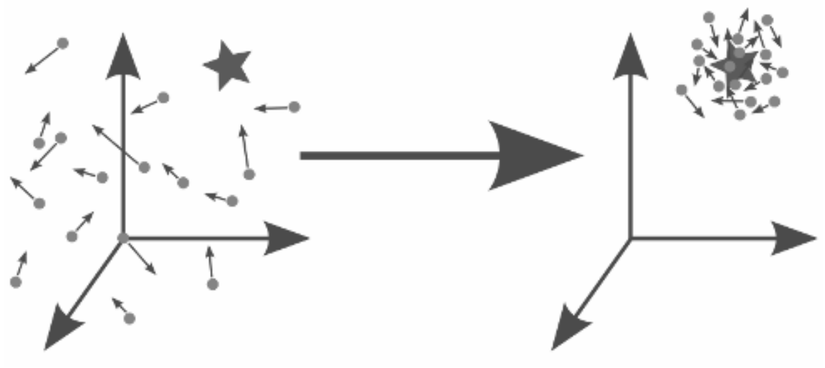
$$\begin{cases} V_{t+1} = \omega * V_t + c_1 r_1 (X_{pbest} - X_t) + c_2 r_2 (X_{gbest} - X_t) \\ X_{t+1} = X_t + V_{t+1} \end{cases}$$

Cognitive Coefficient

Social Coefficient

Updated Position

Previous Position



https://esa.github.io/pagmo2/docs/cpp/algorithms/pso.html

Key Idea I: EcoLife's Dynamic PSO

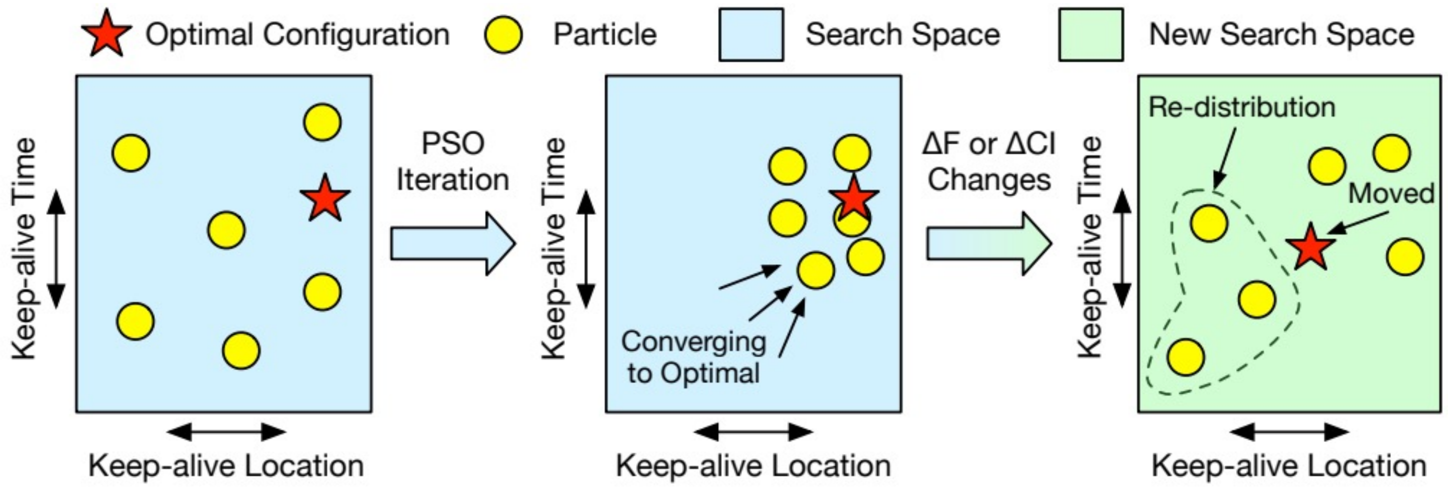
Dynamically adjust these weights based on the changes in carbon intensity and function invocation.

Absolute Changes of Function Invocations

Absolute Changes of Carbon Intensity

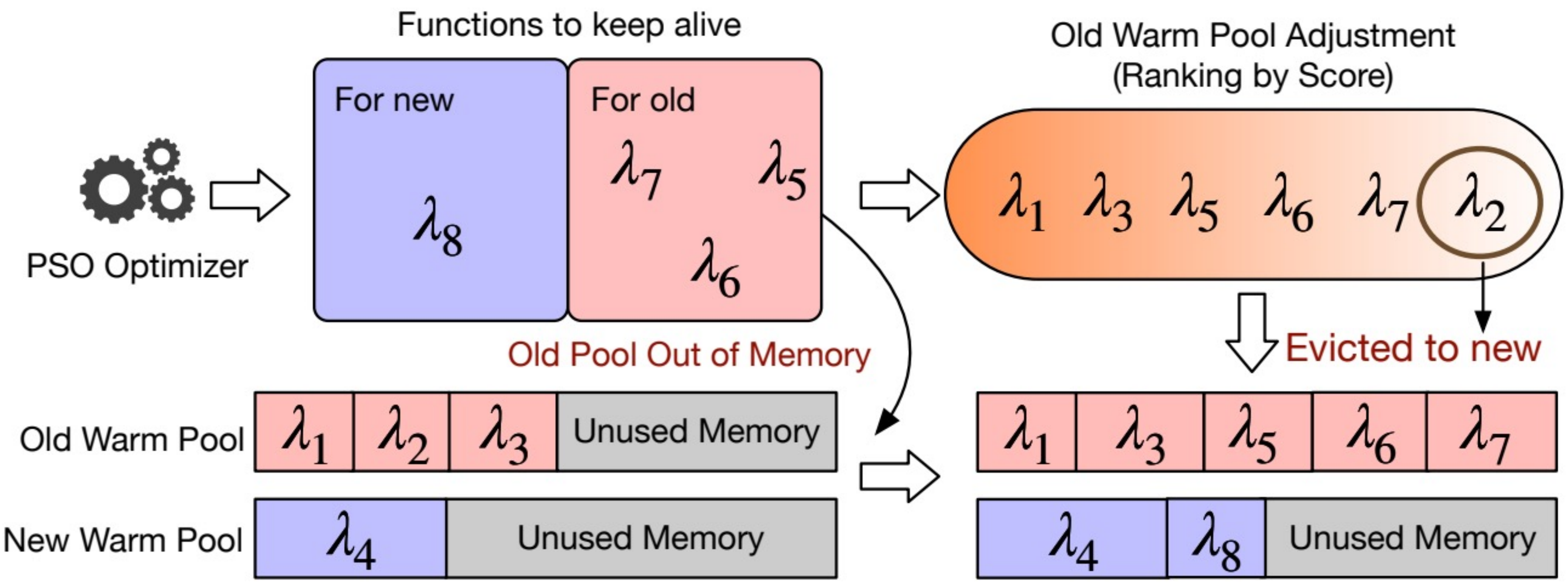
$$w = w_{\max}\left(\frac{\Delta F}{\Delta F_{\max}} + \frac{\Delta \mathrm{CI}}{\Delta \mathrm{CI}_{\max}}\right)$$

$$c_1 = c_2 = c_{\max}\left(1 - \frac{\Delta F}{\Delta F_{\max}} - \frac{\Delta \mathrm{CI}}{\Delta \mathrm{CI}_{\max}}\right)$$



Perception-Response Mechanism: EcoLife detects variations and divides the particle swarm into two halves.

Key Idea II: EcoLife's Warm Pool Adjustment

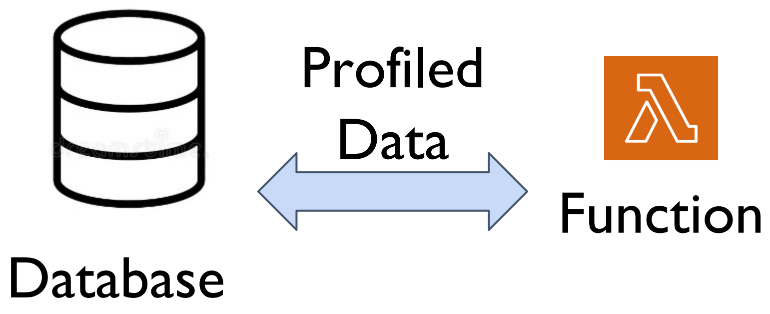


EcoLife adopts a priority eviction mechanism to sort functions already kept alive in the warm pool as well as those about to be kept alive to find the best arrangement.

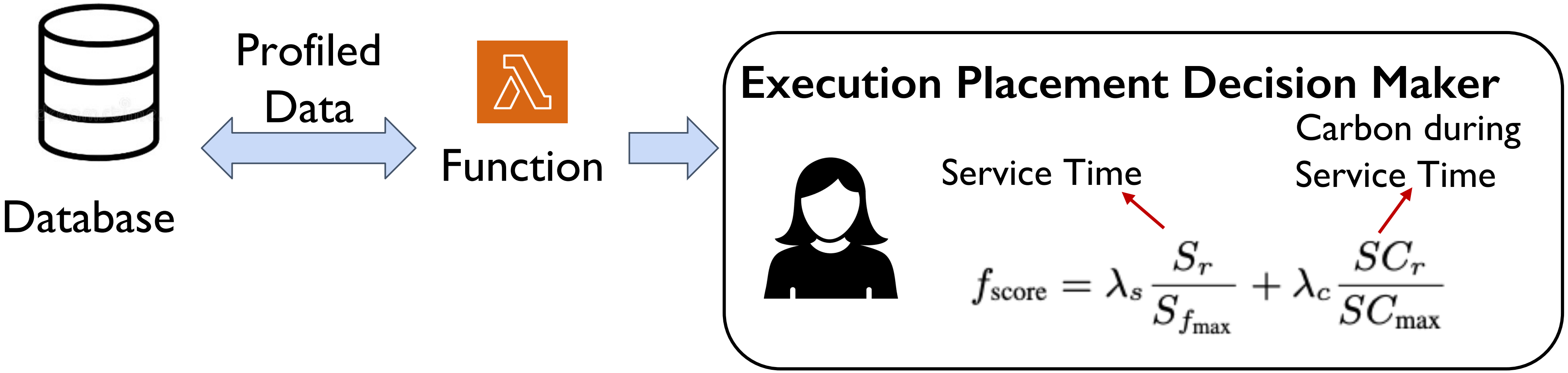
EcoLife Optimization Workflow

Function

EcoLife Optimization Workflow

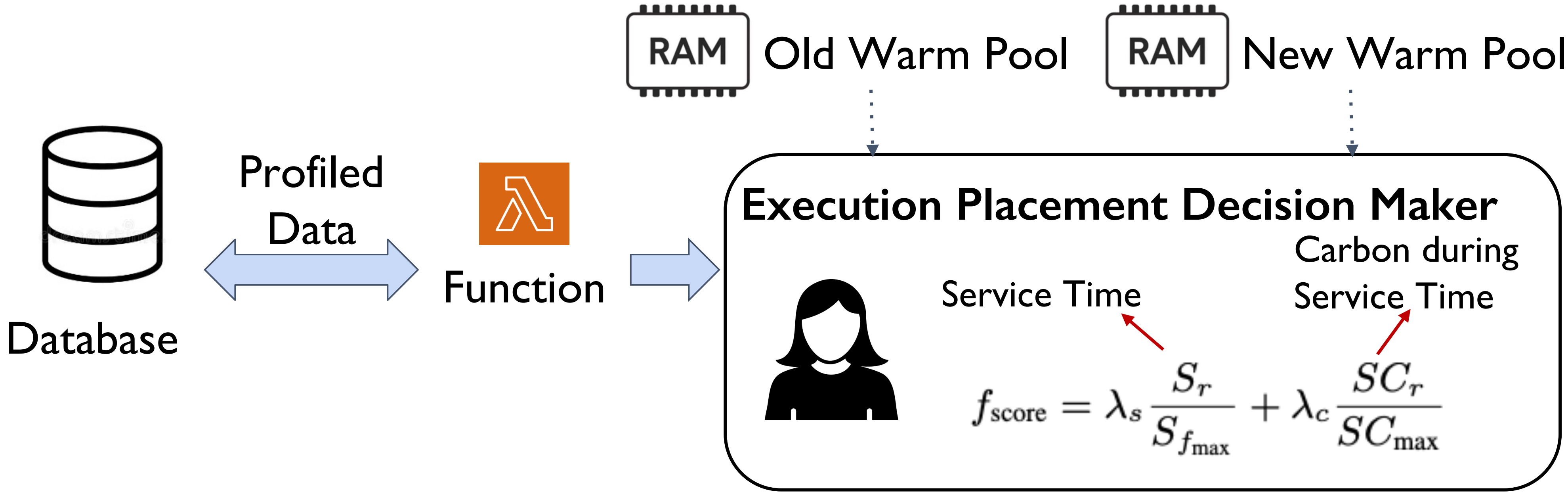


EcoLife Optimization Workflow

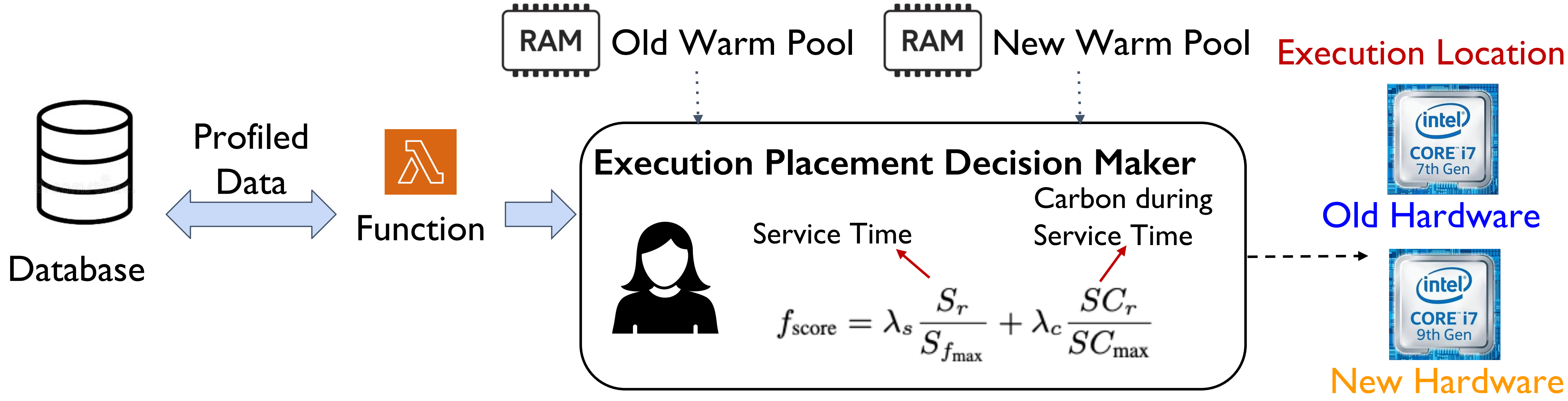


$$f_{\text{score}} = \lambda_s \frac{S_r}{S_{f_{\max}}} + \lambda_c \frac{SC_r}{SC_{\max}}$$

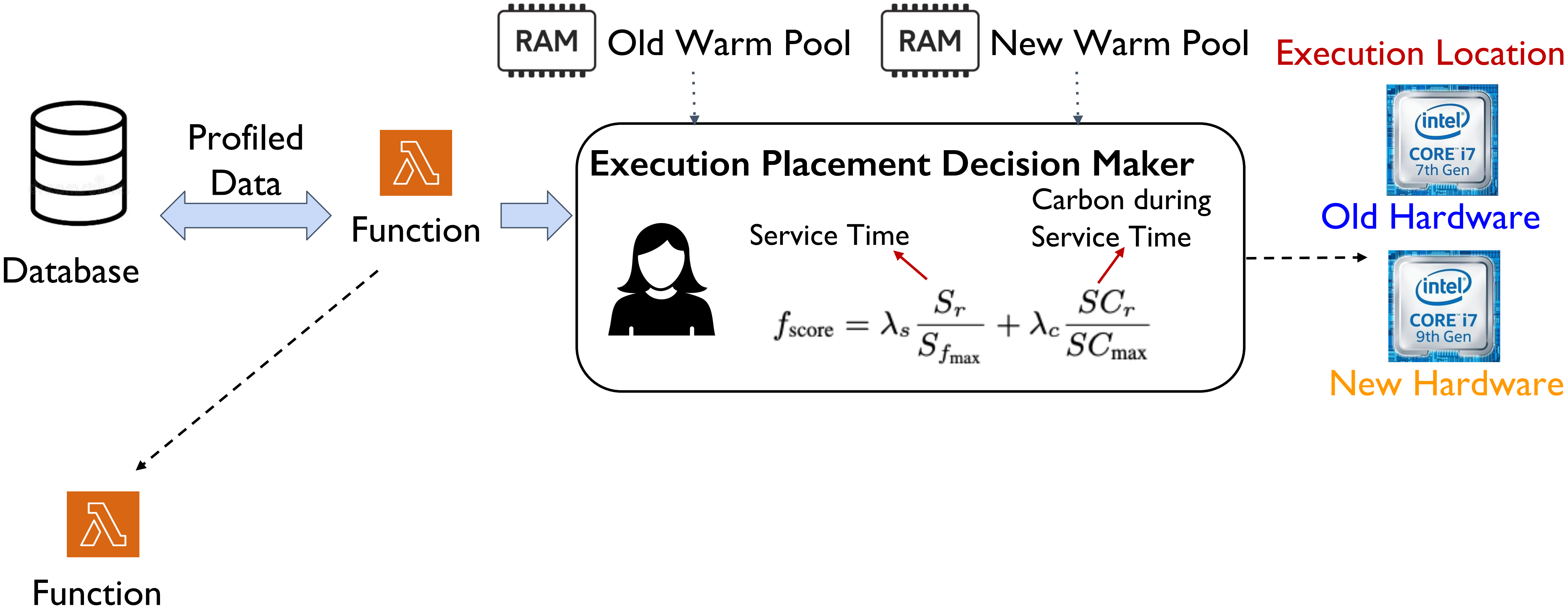
EcoLife Optimization Workflow



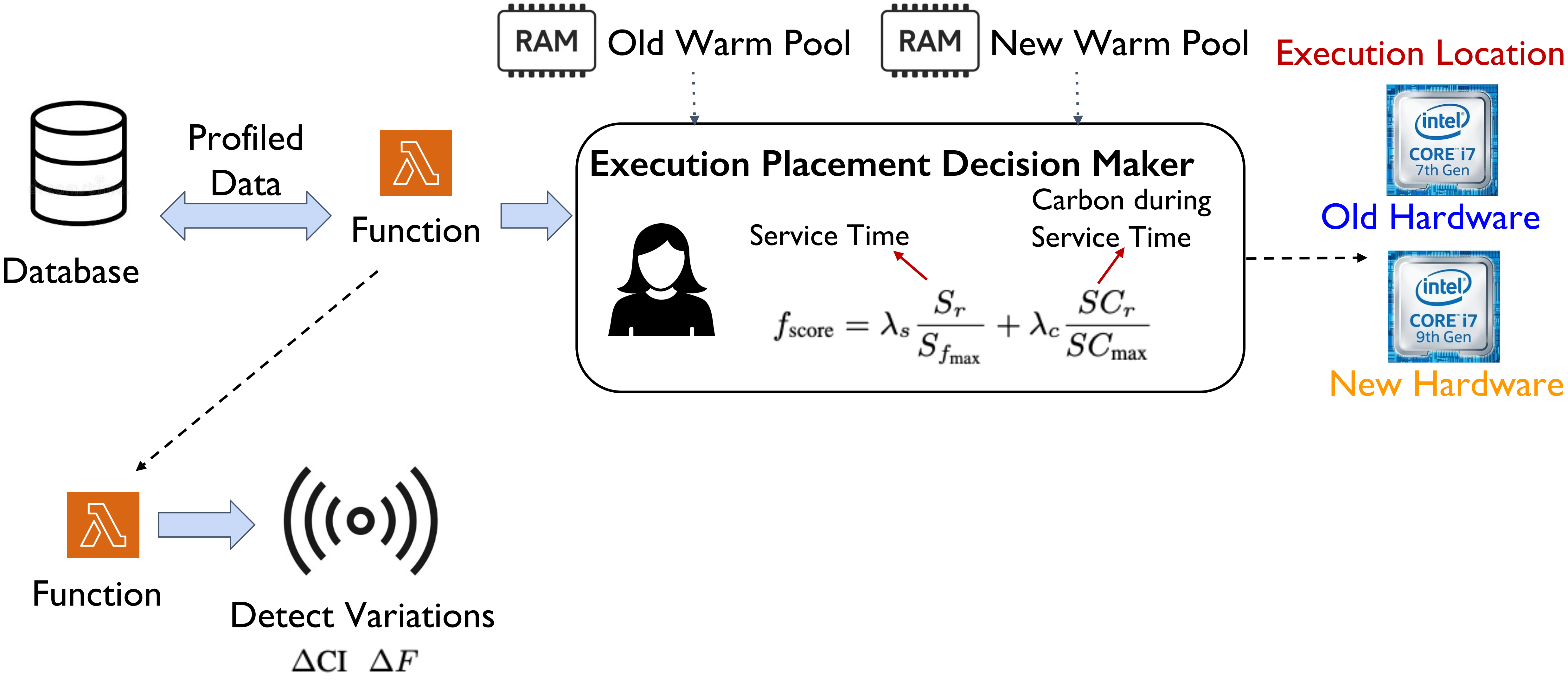
EcoLife Optimization Workflow



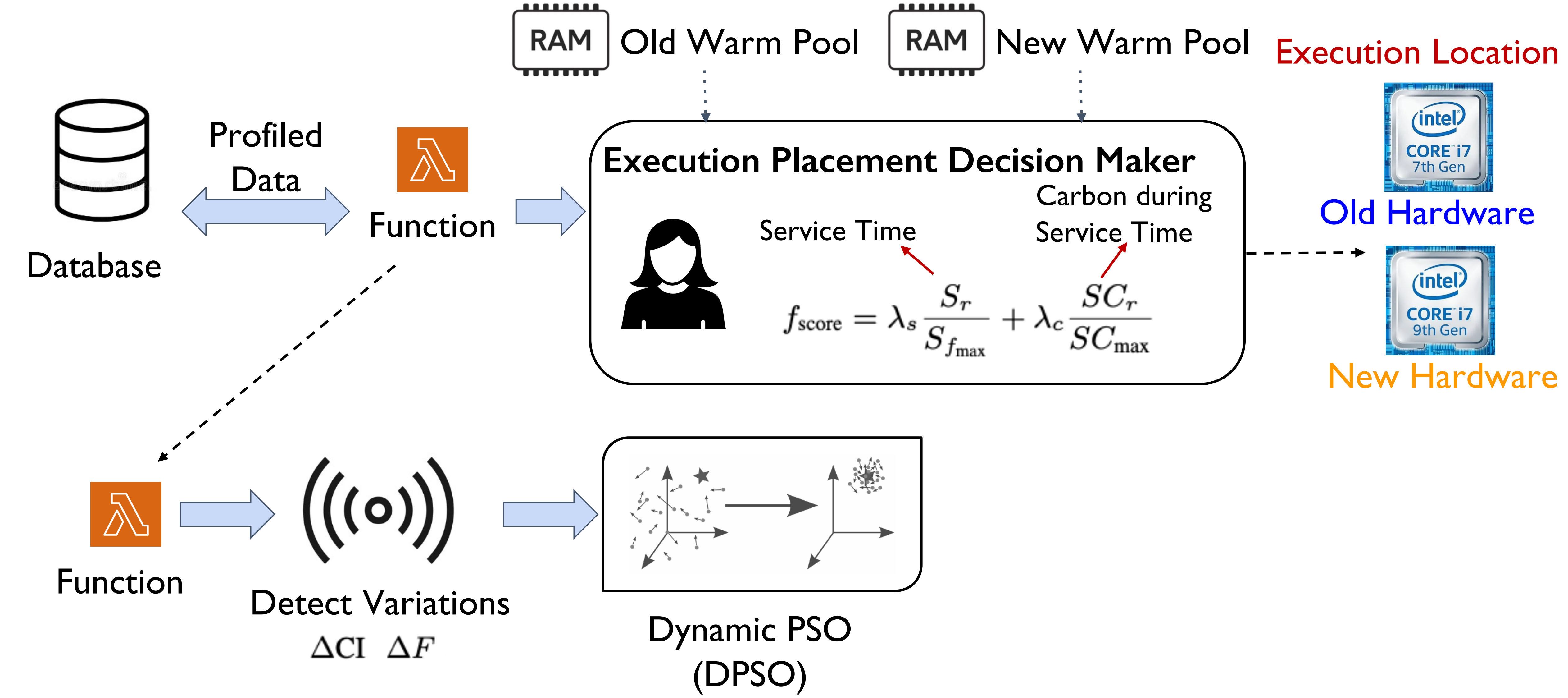
EcoLife Optimization Workflow



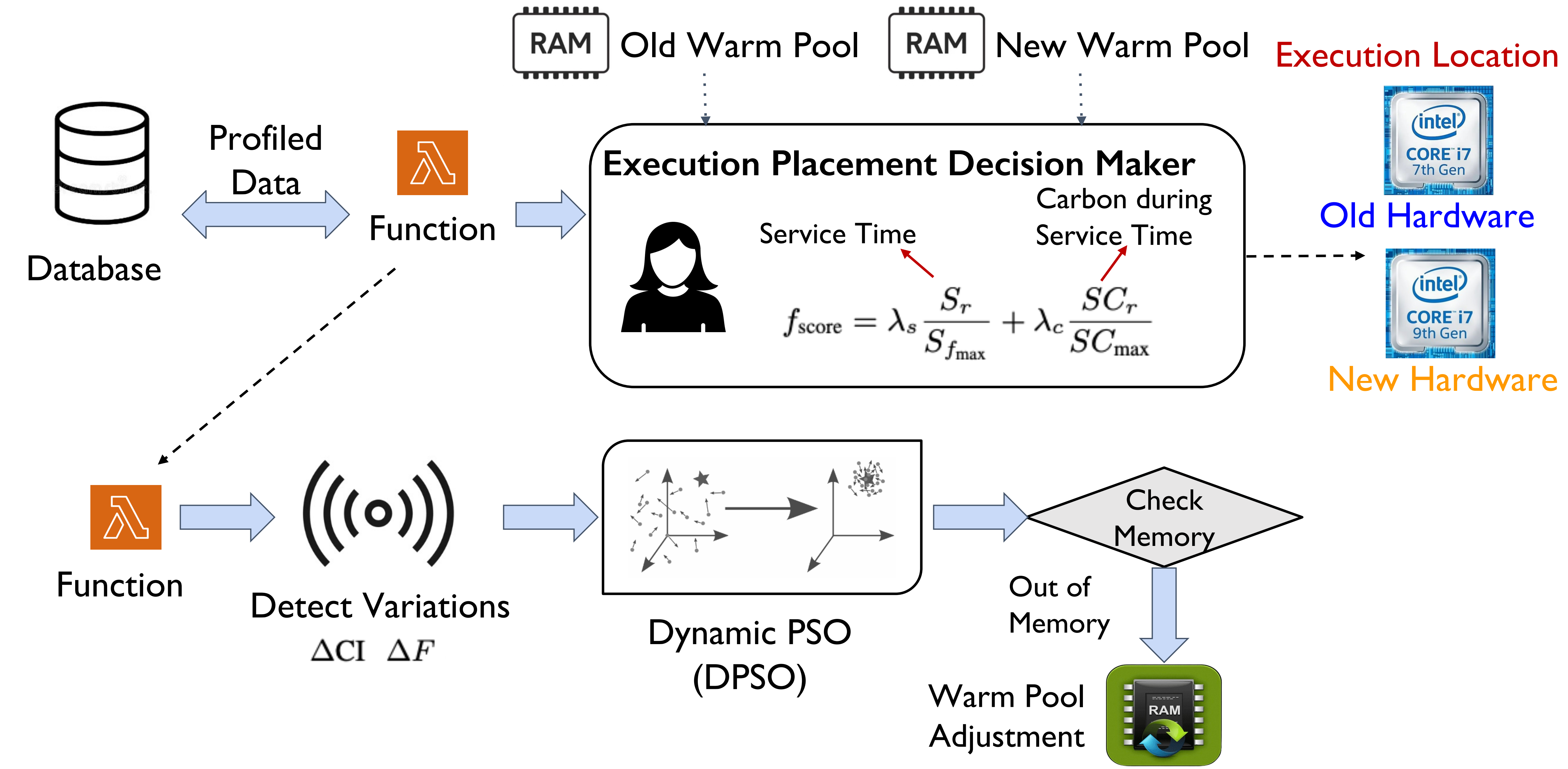
EcoLife Optimization Workflow



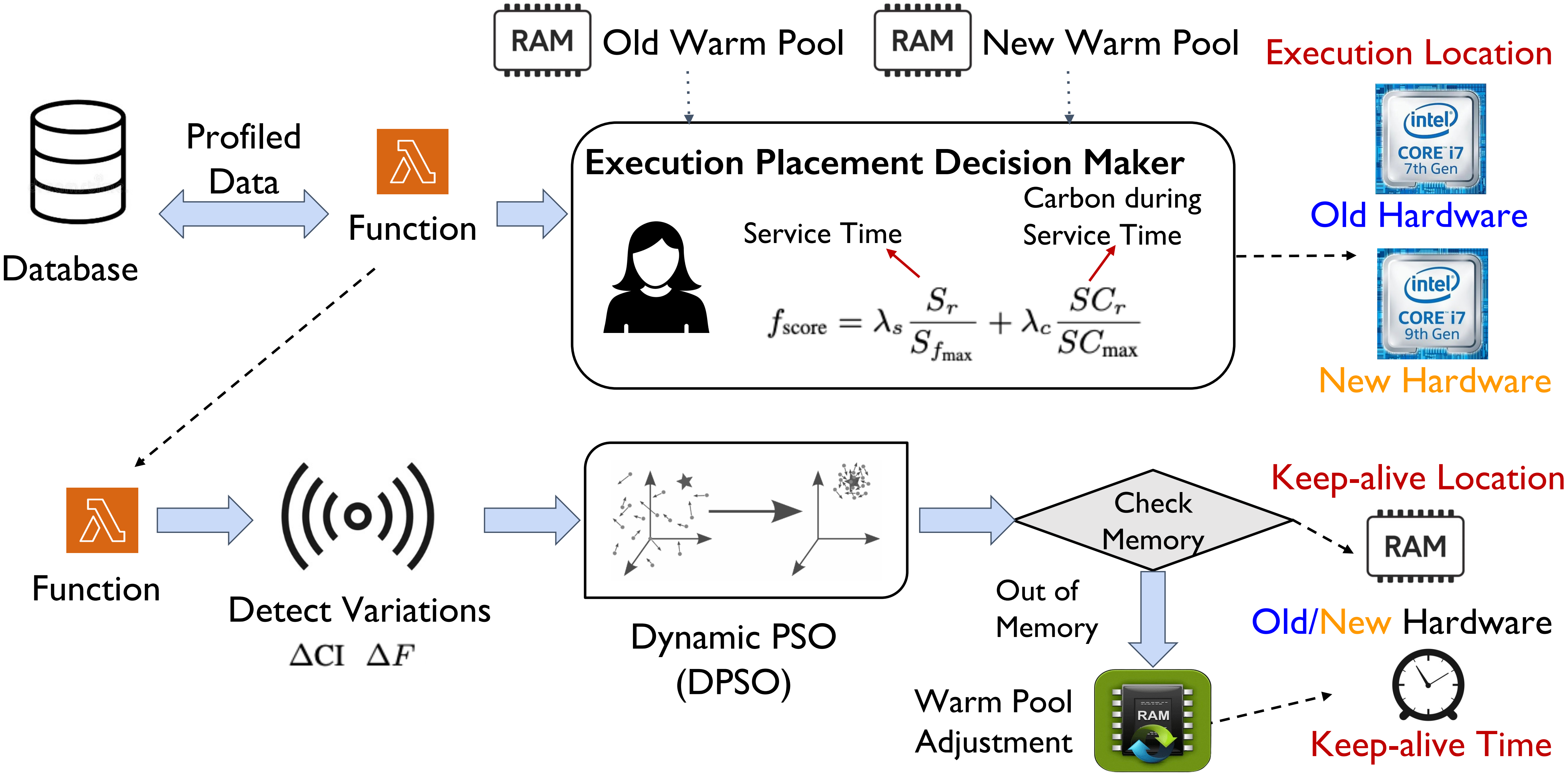
EcoLife Optimization Workflow

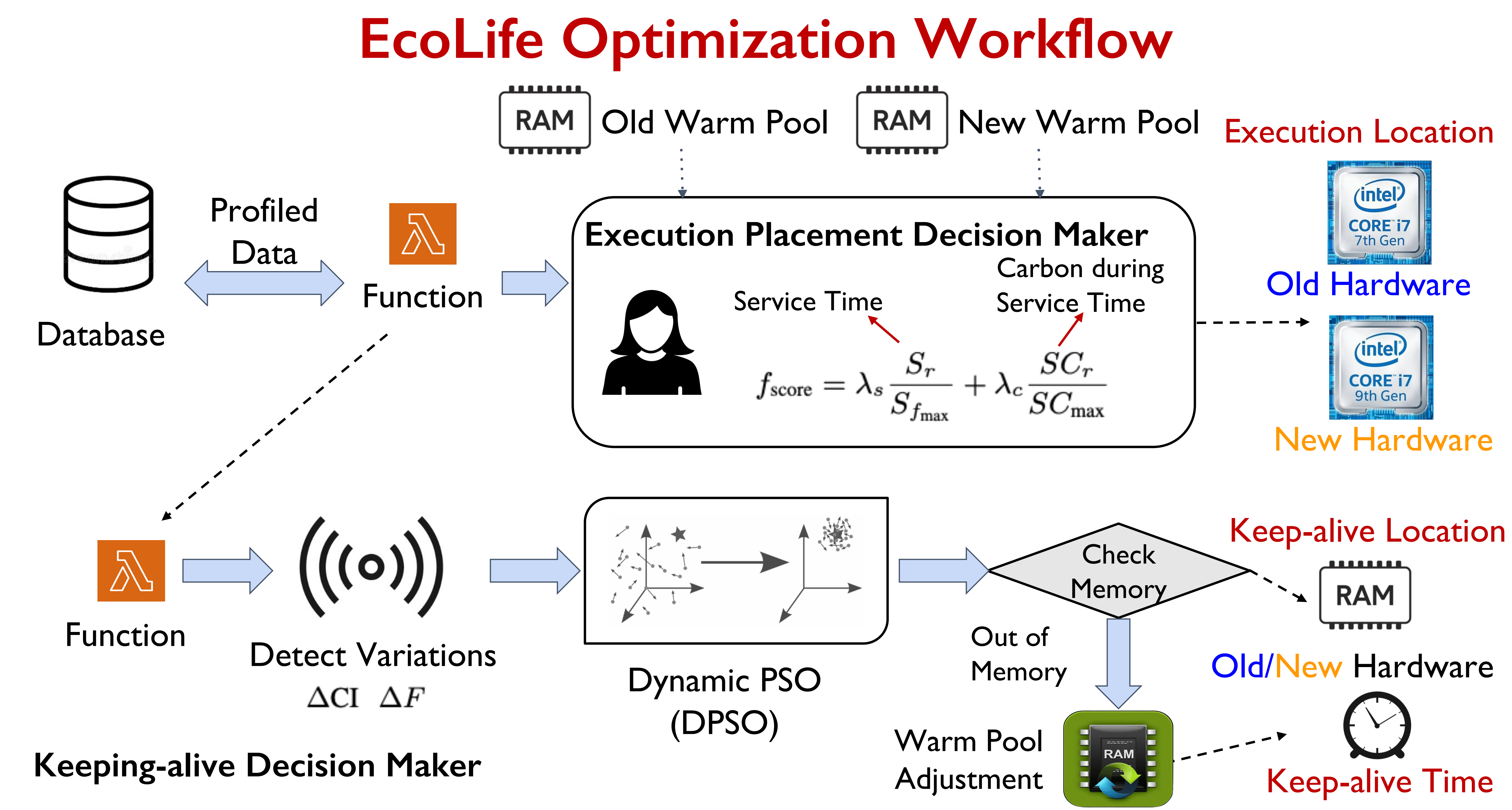


EcoLife Optimization Workflow



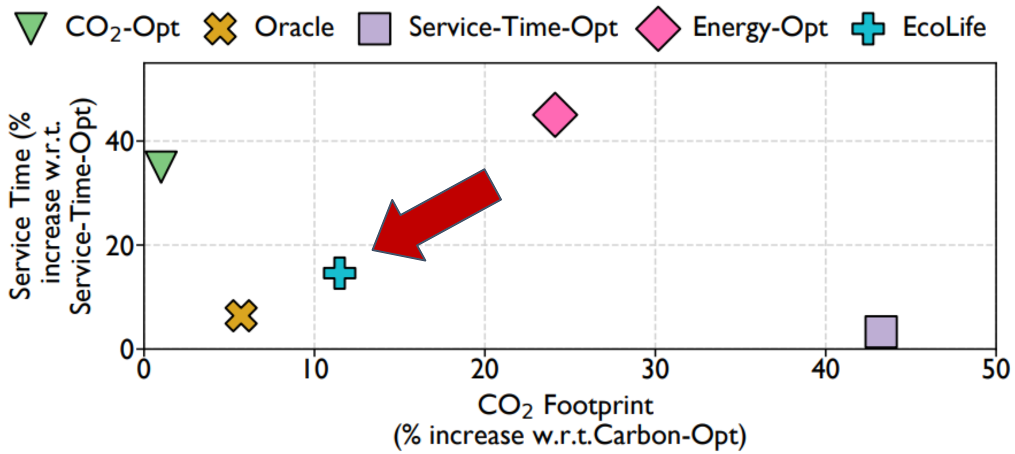
EcoLife Optimization Workflow

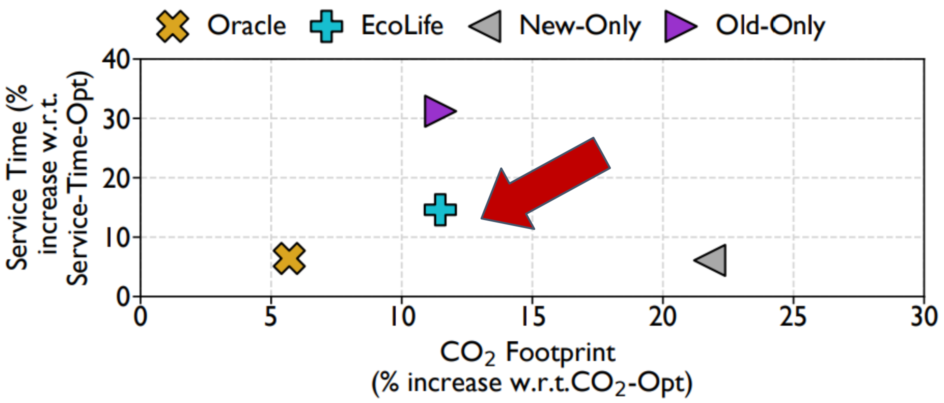



EcoLife Optimization Workflow
RAM
Old Warm Pool
RAM
New Warm Pool
Execution Location
Database
Profiled Data
Function
Execution Placement Decision Maker
Service Time
Carbon during Service Time
$f_{\text{score}} = \lambda_s \frac{S_r}{S_{f_{\max}}} + \lambda_c \frac{SC_r}{SC_{\max}}$
intel CORE i7 7th Gen
Old Hardware
intel CORE i7 9th Gen
New Hardware
Function
Detect Variations
ΔCI ΔF
Dynamic PSO (DPSO)
Check Memory
Out of Memory
Keep-alive Location
RAM
Old/New Hardware
Warm Pool Adjustment
RAM
Keep-alive Time
Keeping-alive Decision Maker

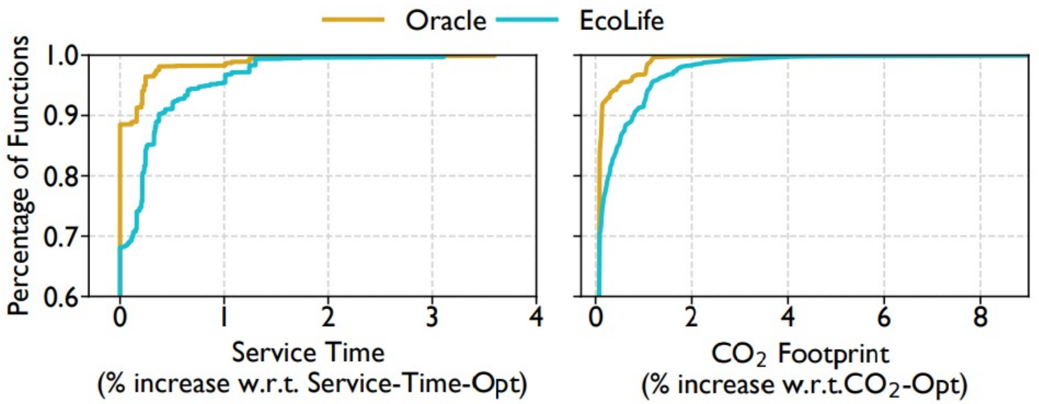
EcoLife is Effective at Reducing Both Carbon Footprint and Service Time

EcoLife provides close to Oracle savings in both carbon footprint and service time.





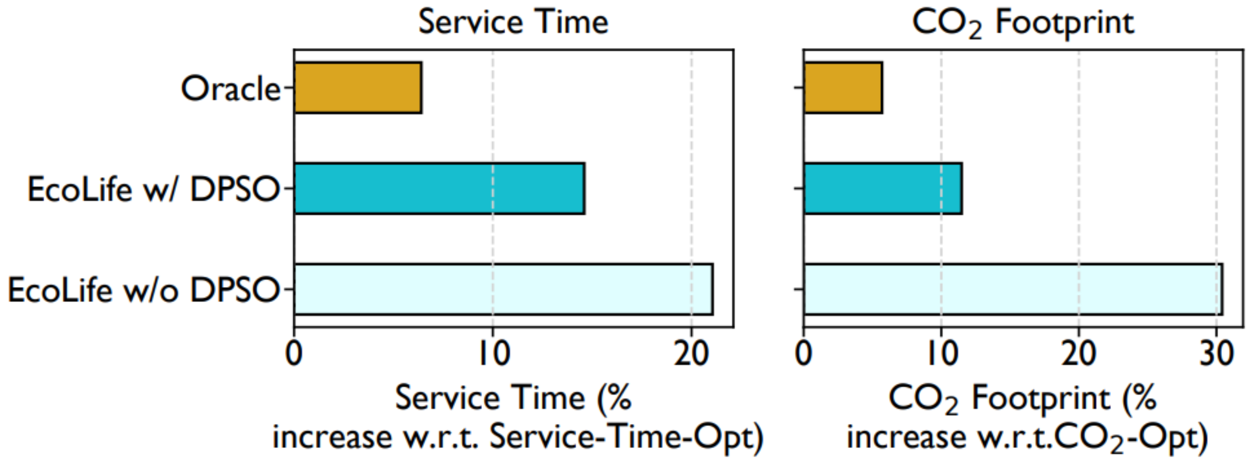
EcoLife outperforms single-generation only solutions.



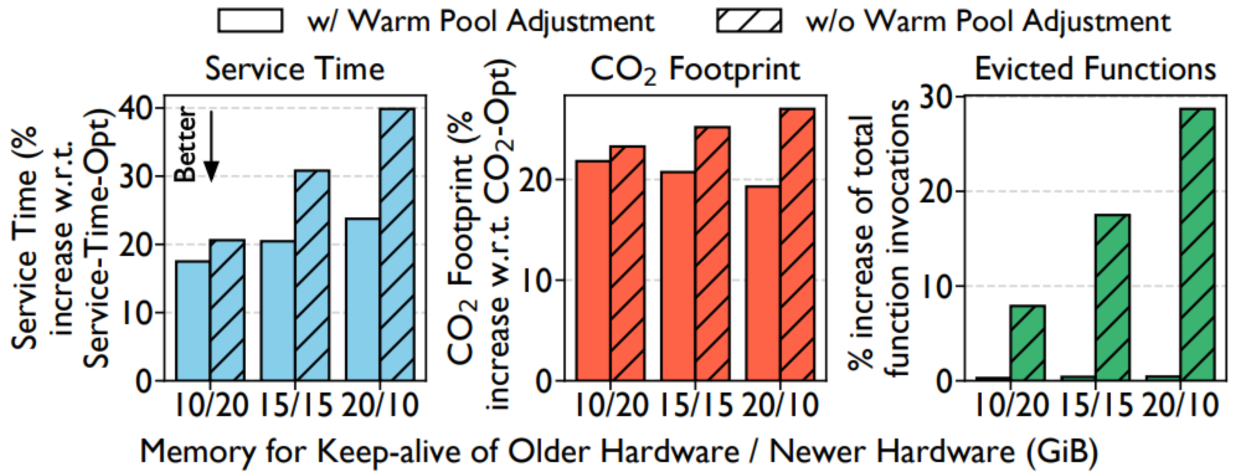
EcoLife stays close to the Oracle for different types of functions.

Novel PSO Extensions of EcoLife are Key to its Efficacy

Dynamic PSO (DPSO) improves the effectiveness of EcoLife, especially in the terms of carbon footprint.

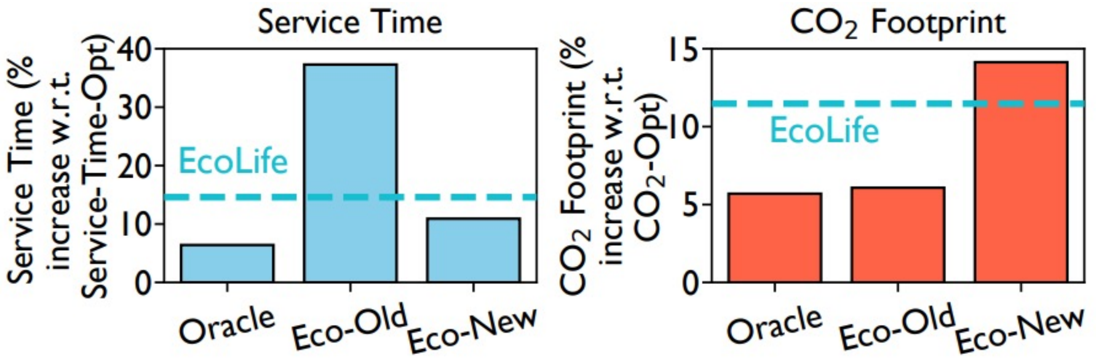


Warm Pool Adjustment strategy increases the hardware utilization, reducing service time and carbon footprint.

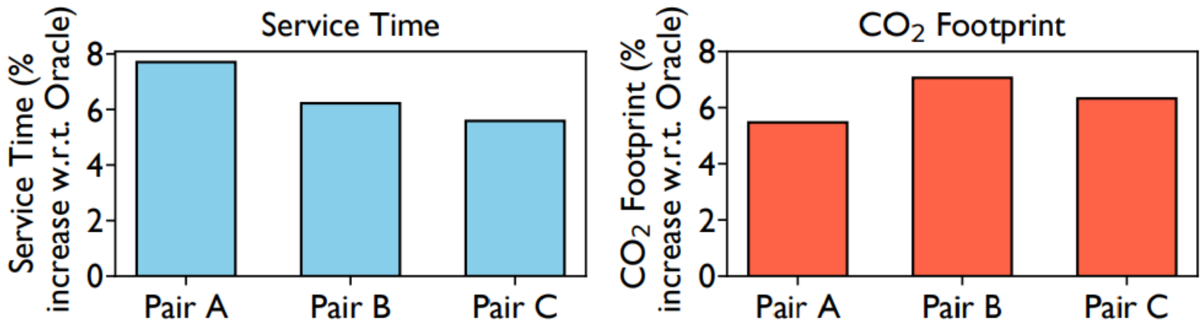


EcoLife is Robust under Different Operating Scenarios

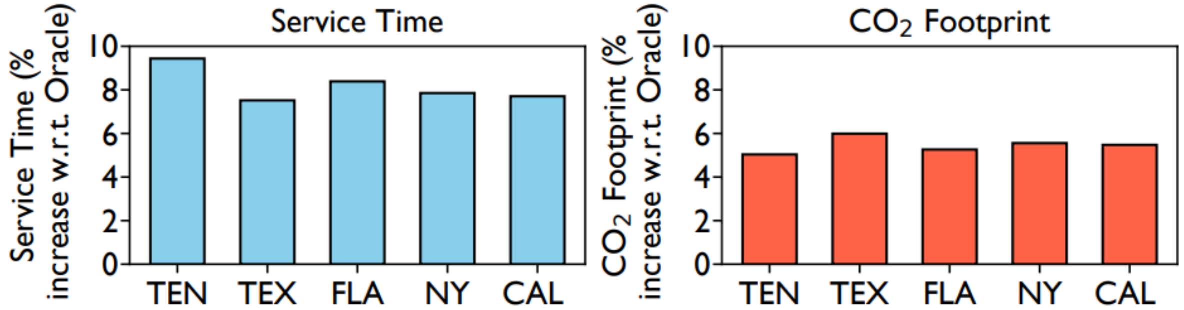
EcoLife can be applied to single-generation hardware as well.



EcoLife is effective and close to Oracle across different multi-generation hardware pairs.



EcoLife is effective and close to Oracle across different geographical regions.



Summary of Key Contributions

- ✓ EcoLife is the first scheduler that reduces the carbon footprint of serverless functions.
- ✓ EcoLife introduces novel key ideas to effectively leverage Particle Swarm Optimization (PSO) in the context of serverless scheduling and sustainability.
- ✓ EcoLife is consistently within 7.7% and 5.5% points from Oracle in terms of service time and carbon footprint.

EcoLife

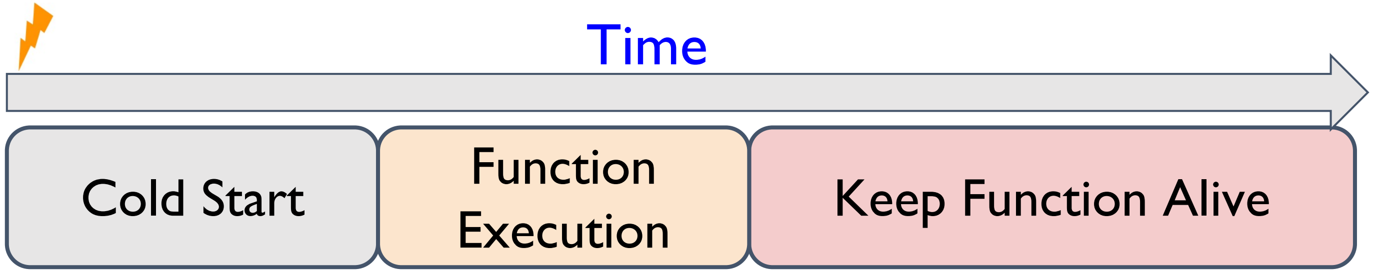
Contact
Yankai Jiang
jiang.yank@northeastern.edu

Bonus Slides

Contact

Yankai Jiang
jiang.yank@northeastern.edu

Carbon Footprint for Serverless Computing



$$\text{Operational } CO_2 = \text{Energy} \times \text{CI}$$

$$\text{Embodied } CO_2 = \frac{\text{Time}}{\text{LifeTime}} \times \text{Embodied}_{\text{Hardware}}$$

CPU

Energy during Service (include Cold Start)

Energy during Keep-alive

$$\text{CPU Operational } CO_2 = \left(E_{\text{CPU}}^{\text{Service}} + \frac{E_{\text{CPU}}^{\text{Keep-alive}}}{\text{Core}_{\text{num}}}\right) \cdot \text{CI}$$

Real-time Carbon Intensity

# of CPU Cores

Service Time (include Cold Start)

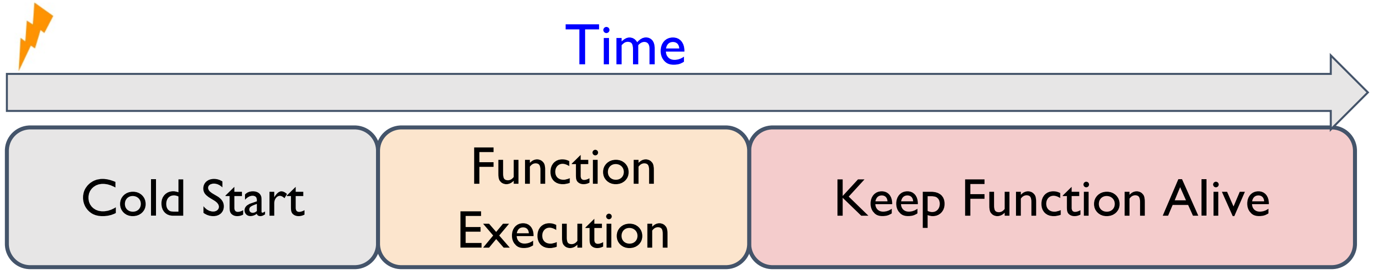
Keep-alive Period

Total Embodied Carbon of CPU

$$\text{CPU Embodied } CO_2 = \frac{S_f}{LT_{\text{CPU}}} \cdot EC_{\text{CPU}} + \frac{k}{LT_{\text{CPU}}} \cdot \frac{EC_{\text{CPU}}}{\text{Core}_{\text{num}}}$$

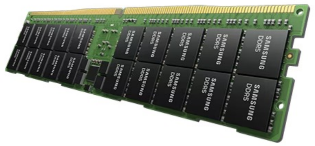
CPU Lifetime

Carbon Footprint for Serverless Computing



$$\text{Operational } CO_2 = \text{Energy} \times \text{CI}$$

$$\text{Embodied } CO_2 = \frac{\text{Time}}{\text{LifeTime}} \times \text{Embodied}_{\text{Hardware}}$$



DRAM

Memory Size of Function → M_f

Energy during Service → $E_{\text{DRAM}}^{\text{Service}}$

Energy during Keep-alive → $E_{\text{DRAM}}^{\text{Keep-alive}}$

$$\text{DRAM Operational } CO_2 = \frac{M_f}{M_{\text{DRAM}}} \cdot (E_{\text{DRAM}}^{\text{Service}} + E_{\text{DRAM}}^{\text{Keep-alive}}) \cdot \text{CI}$$

Memory Size of DRAM → M_{DRAM}

Real-time Carbon Intensity → CI

Service Time (include Cold Start) + Keep-alive Time → $S_f + k$

$$\text{DRAM Embodied } CO_2 = \frac{S_f + k}{LT_{\text{DRAM}}} \cdot \frac{M_f}{M_{\text{DRAM}}} \cdot EC_{\text{DRAM}}$$

DRAM Lifetime → LT_{DRAM}

Total Embodied Carbon of DRAM → EC_{DRAM}

Why does EcoLife Use PSO?

- PSO can rapidly converge to global optima due to its exploration-exploitation balance. (Low decision-making overhead)
- PSO can continuously adapt to changing conditions and provide near-optimal solutions in dynamic environments, which is needed in a serverless context.
- PSO can reduce the carbon footprint by 17.4% and service time by 7.2%, compared to Genetic Algorithm.
- PSO can have 6.2% reduction in carbon footprint and a 13.46% decrease in service time compared to the Simulated Annealing Algorithm.