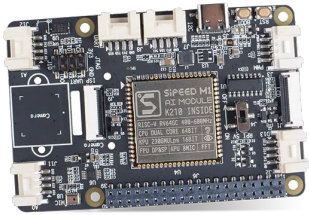
fReeLoaders: An IoT Ecosystem for Real-Time Deadline-Driven Task Scheduling using Reinforcement Learning

Marshall Clyburn*, University of Virginia
Nabeel Nasir*, UC Santa Barbara
Md Fazlay Rabbi Masum Billah, Amazon.com
Victor Ariel Leal Sobral, University of Virginia
Jiechao Gao, Stanford University
Fateme Nikseresht, University of Virginia
Brad Campbell, University of Virginia

IoT applications are advancing

Cognitive assistance
in AR headsets

Object detection
in home security

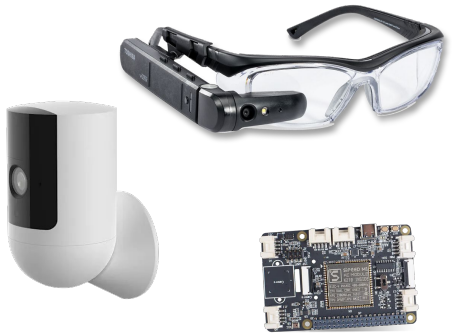


Audio classification
in low-power systems

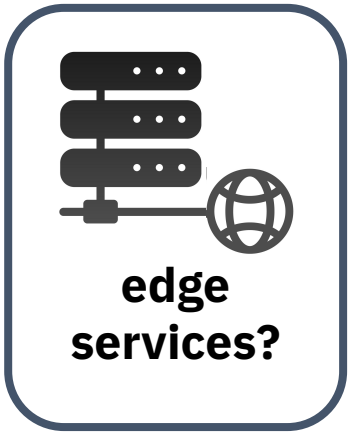
Growing application compute demand.

Drive for lower power end devices.

The IoT relies on external compute



compute-intensive workloads



Edge service growth is limited

EdgeMicro

EdgeMicro is no longer active.

EDGENOMICS

Ericsson's Edge Gravity drops out of favour

By Ray Le Maistre

Jun 15, 2020

- Global edge cloud unit has been closed
- Internal startup was less than two years old
- Example of how tough it is to compete with the webscale giants

Edge service growth is limited

Crown Castle scales back small cell build out, outlines plans to connect metro data center markets

Planned cancelation of 7,000 small cells will save $800m in future capital spend

October 18, 2024 By: Paul Lipscombe Have your say

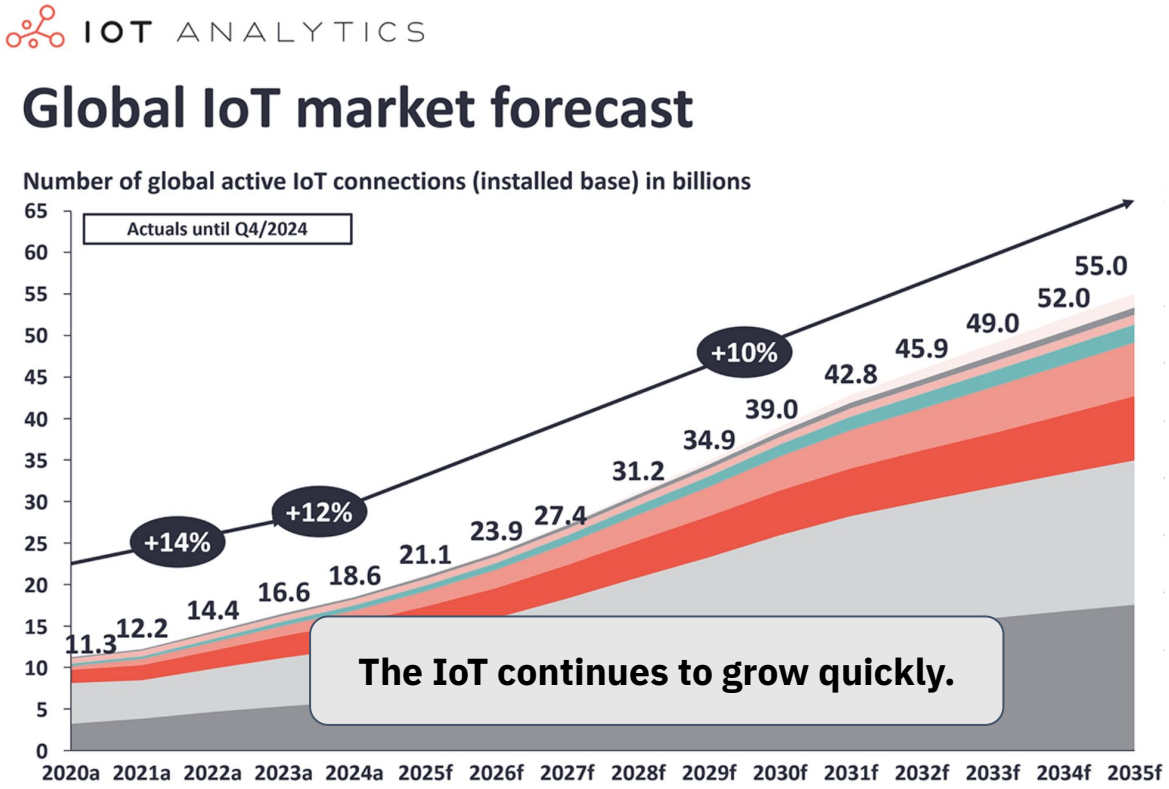
Verizon admits to miscalculations on 5G, edge computing and private networks

'The mobile edge compute and private 5G networks ... the adoption curve [is] a little slower than maybe we would like,' admitted Verizon CFO Matt Ellis during the company's earnings conference call.

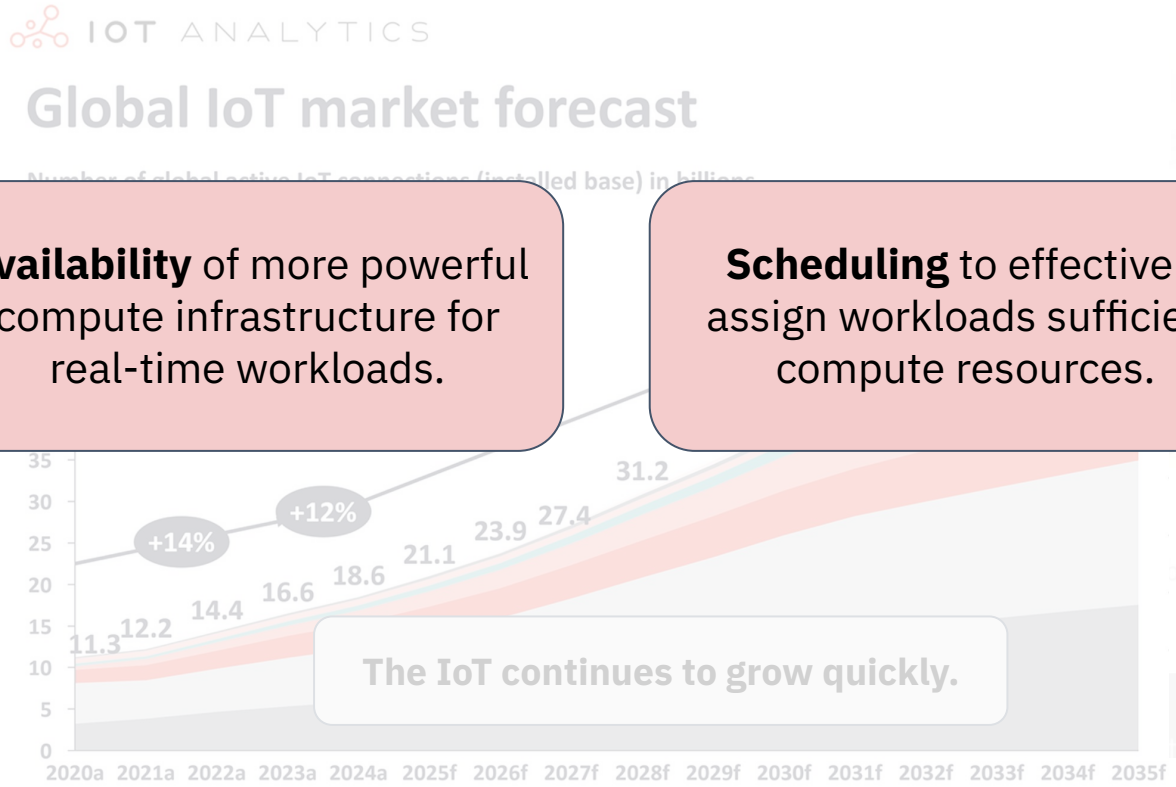
Mike Dano, Editorial Director, 5G & Mobile Strategies, Light Reading
January 24, 2023

5 Min Read

Edge service growth is limited



Edge service growth is limited

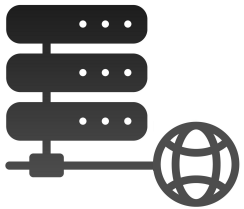


Availability of more powerful compute infrastructure for real-time workloads.

Scheduling to effectively assign workloads sufficient compute resources.

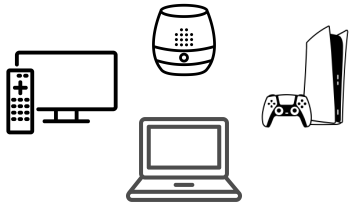
Hyper-local compute provides **availability**

Edge services



limited deployment

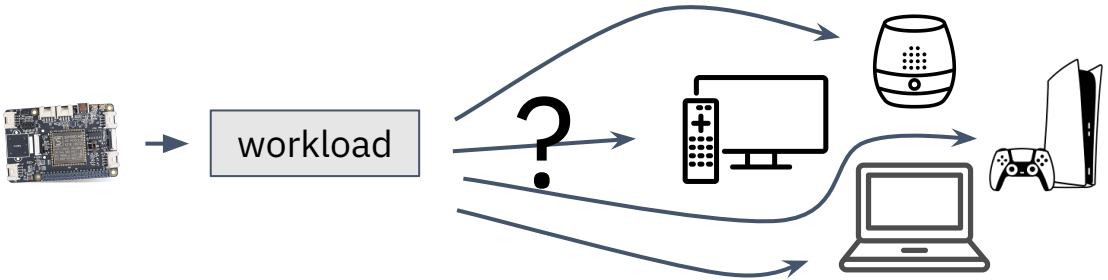
Hyper-local devices



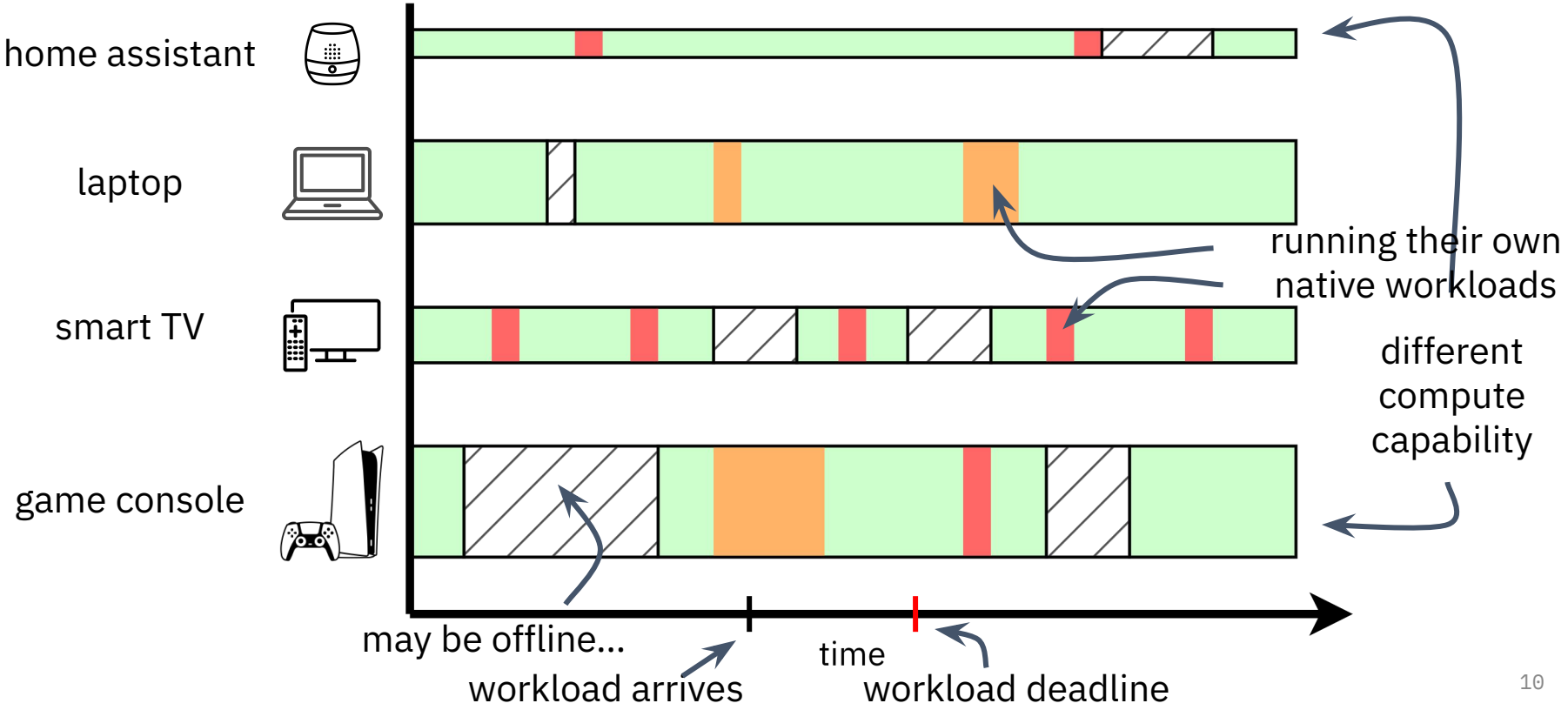
- low latency
- capable compute
- considerable idle time
- stable energy supply
- **high heterogeneity**

Hyper-local poses a scheduling challenge

- **Diverse landscape**: CPU, arch., memory, network, etc.
- Many types of workloads + heterogeneous hosts.
- Good availability, but **must know how to place workloads**.



Scheduling on hyper-local compute



Profiling, a costly approach

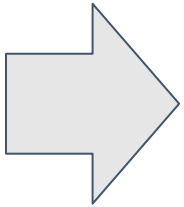
- Predict worst-case execution time.
- But… is time- and energy-intensive…
  - Profiling process? What to capture?
  - Workload, host, environment ➔ intractable.
- Must re-profile (new, updating applications).

Removing *a priori* profiling

Observing workload **deadline satisfaction reveals QoS requirements**, compute capability.

Workloads repeat over time.

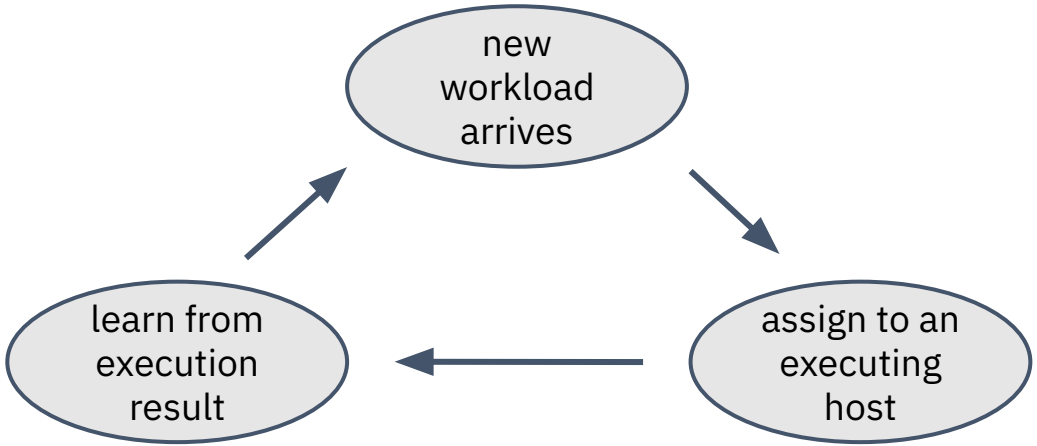
IoT environments are generally **soft real-time systems**.



Experientially learn to meet workload QoS requirements **on-the-fly**.

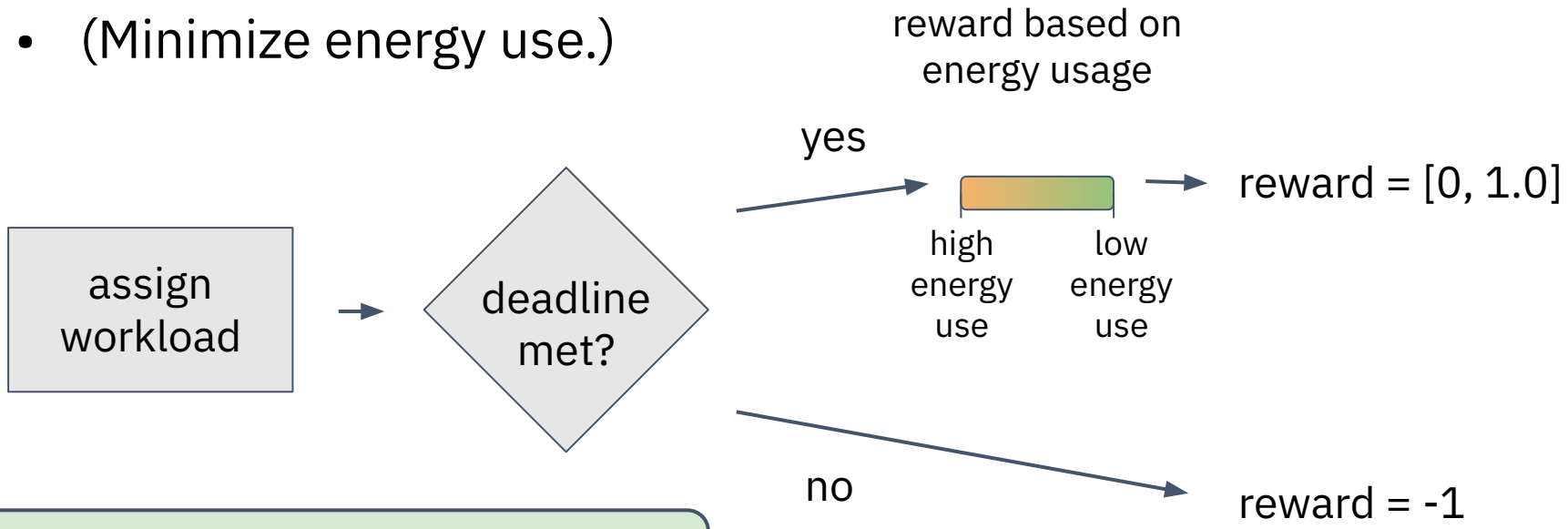
The fReeLoaders scheduler

- Don’t profile, **learn experientially**.
- Trade early performance for adaptability.
- Reinforcement learning for **continuous adaptation**.



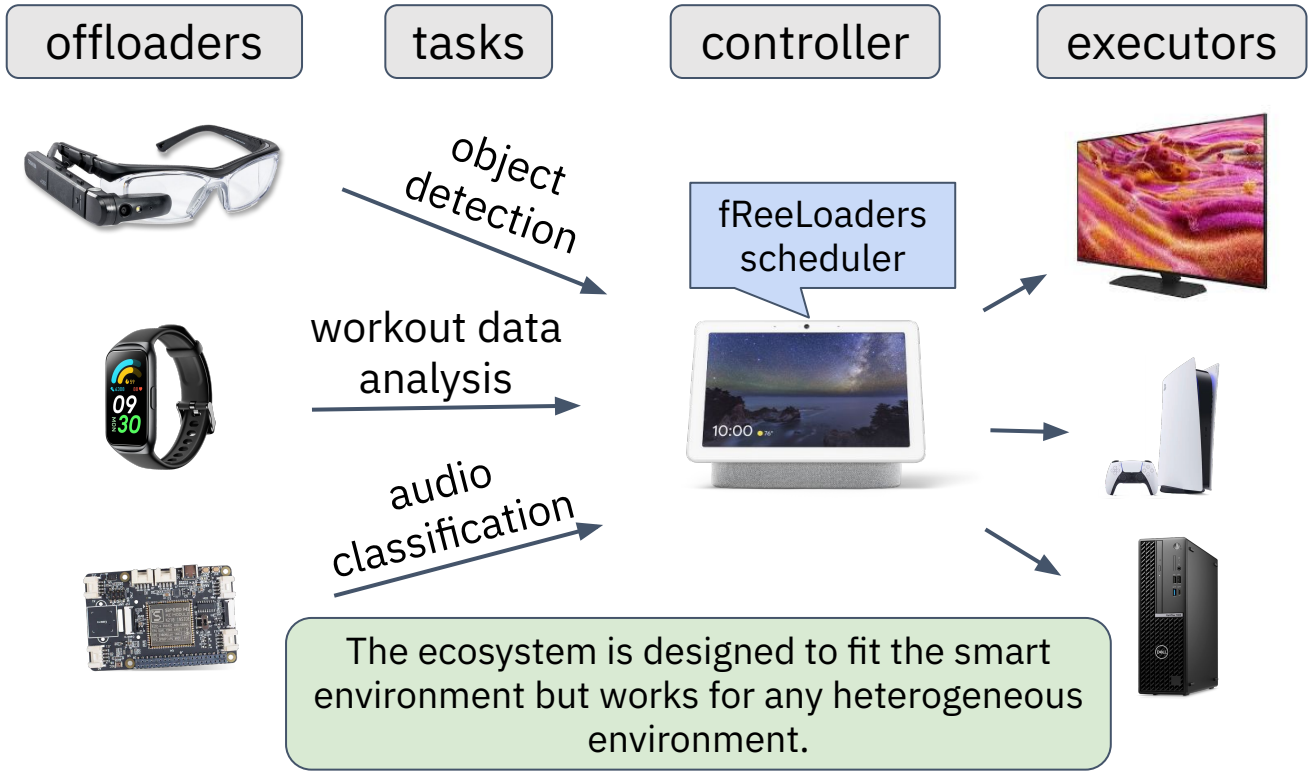
fReeLoaders RL scheduler

- Meet deadlines.
- (Minimize energy use.)

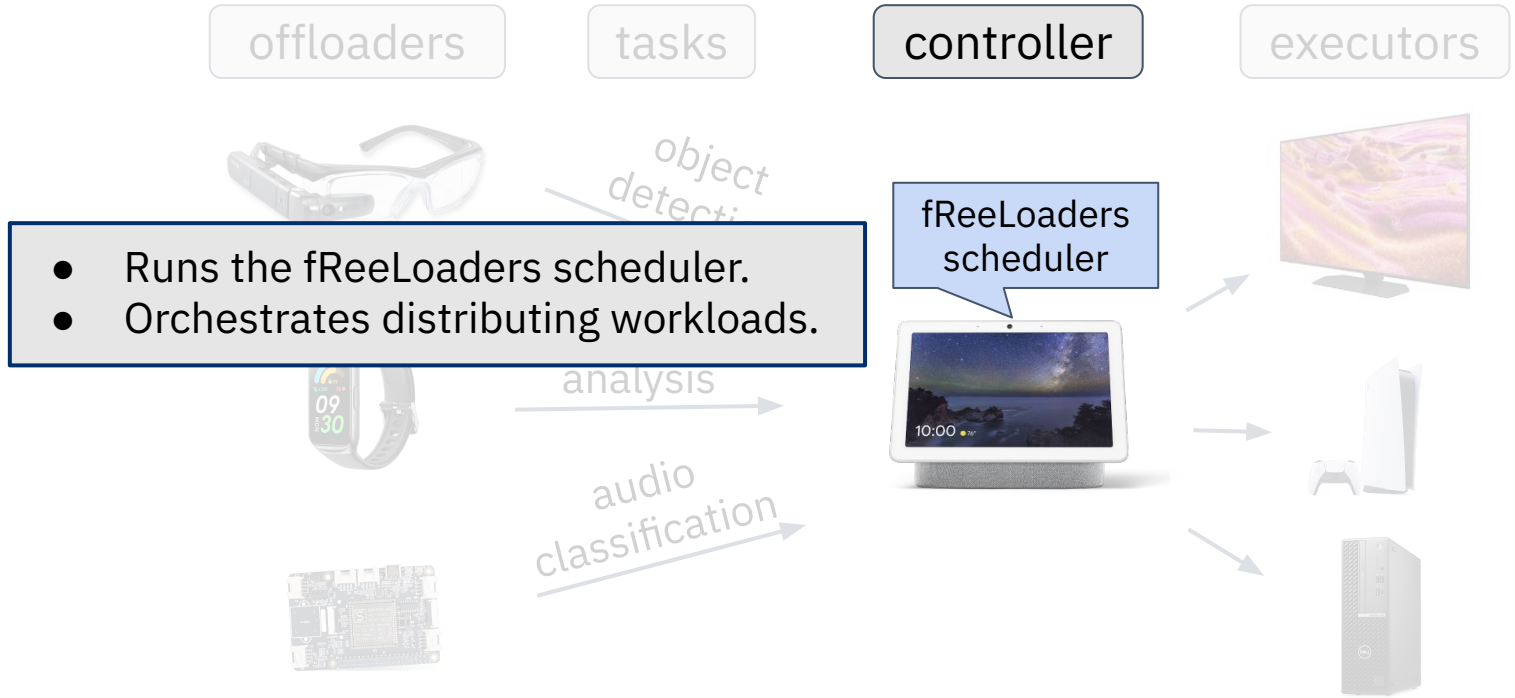


Prioritize meeting deadlines, but secondarily incentivizes driving down energy usage.

The fReeLoaders ecosystem



fReeLoaders: controller



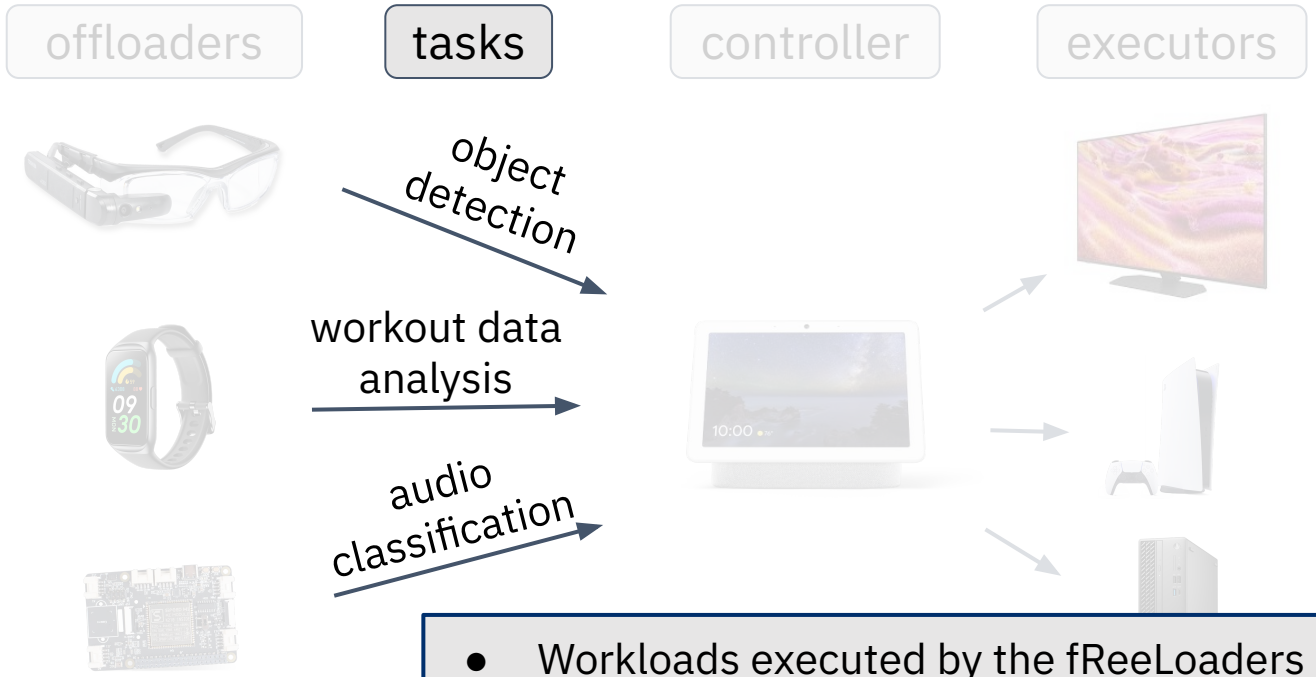
- Runs the fReeLoaders scheduler.
- Orchestrates distributing workloads.

fReeLoaders: offloaders

offloaders | tasks | controller | executors

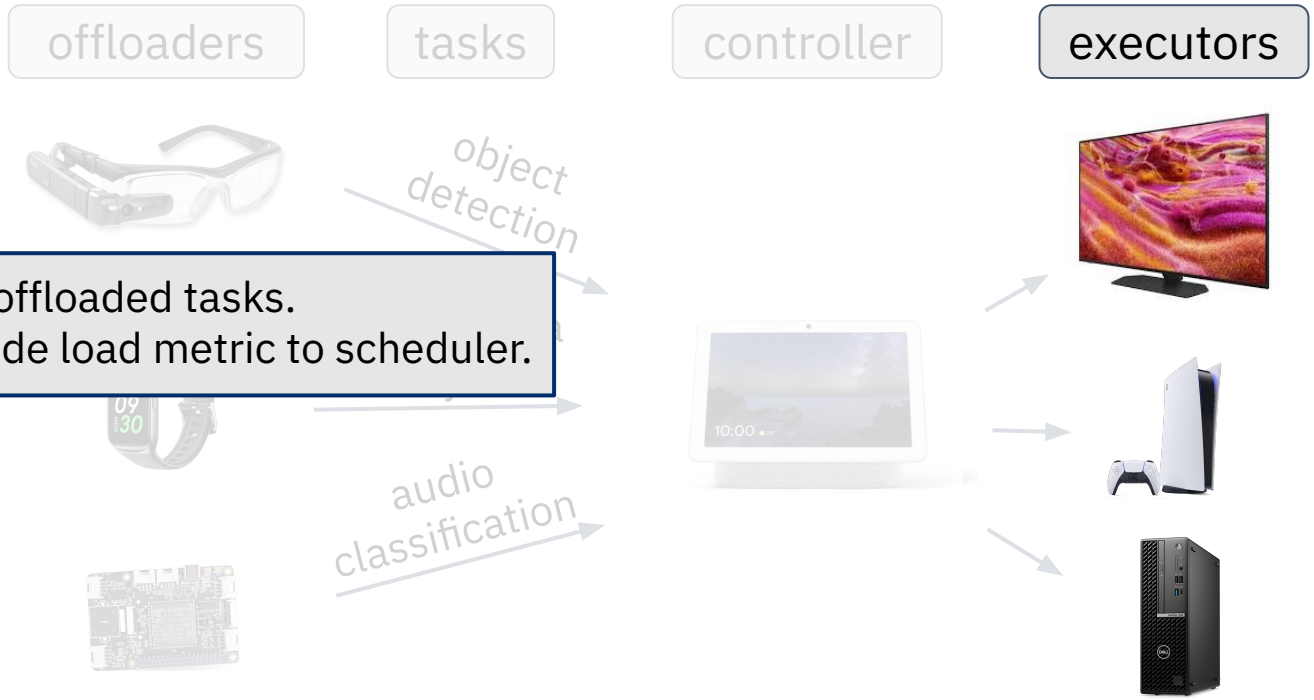
- Compute- or energy-constrained devices.
- Extend or improve applications by offloading.
- Can conserve energy/handle other work after offloading.

fReeLoaders: tasks



- Workloads executed by the fReeLoaders ecosystem.
- Defined by application developer.

fReeLoaders: executors

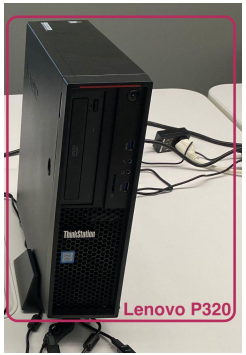


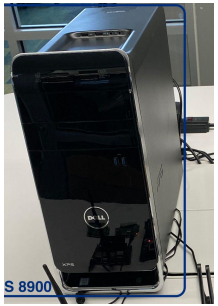
- Run offloaded tasks.
- Provide load metric to scheduler.

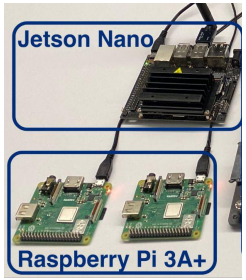
Evaluation hardware

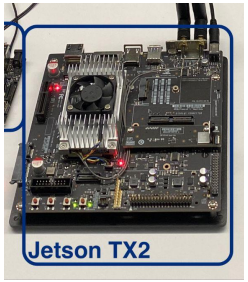
controller

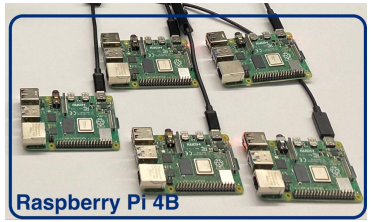
executors











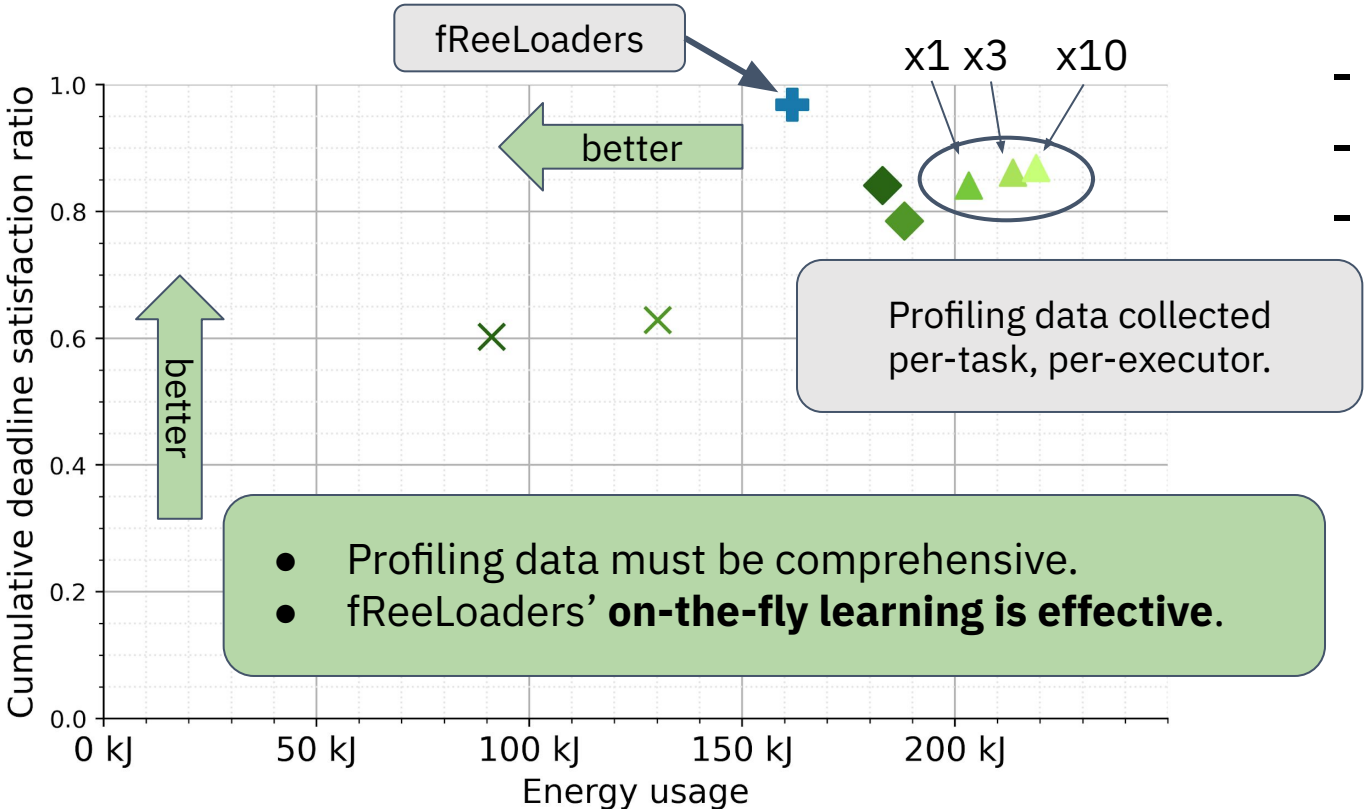
- Ten executors with varied compute capability.
- Specs match common smart devices.

Evaluation tasks

Tasks (10 variants each)	Executor meets deadlines for task?				
	Nano	Pi 3	Pi 4	TX2	PC
loop	✔	✔	✔	✔	✔
matrix multiplication	✘	✘	✔	✔	✔
FFT	✘	**some**	✔	**some**	✔
activity recognition	✘	✘	**some**	✘	✔
object detection	✘	✘	✘	✘	✔
room classification	✔	✔	✔	✔	✔

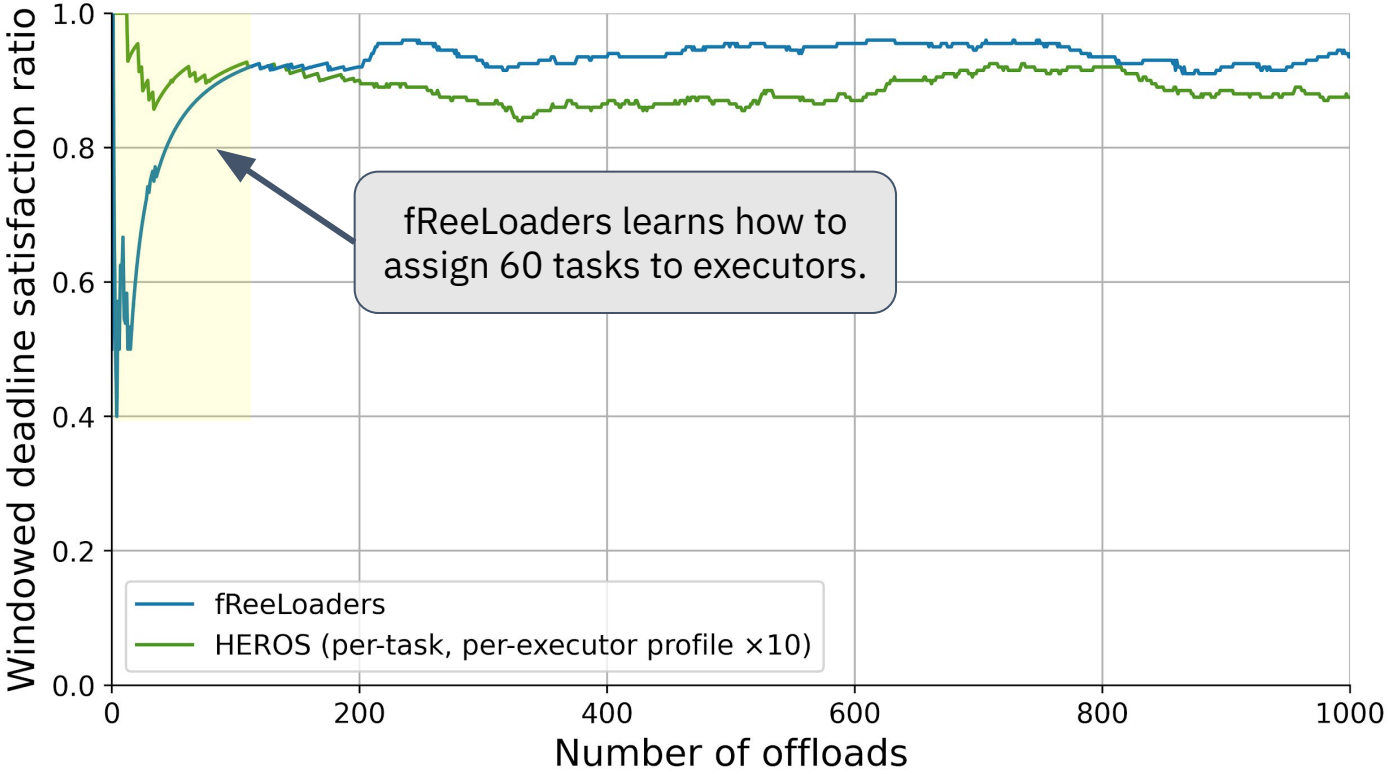
60 different tasks total

A performant alternative to profiling

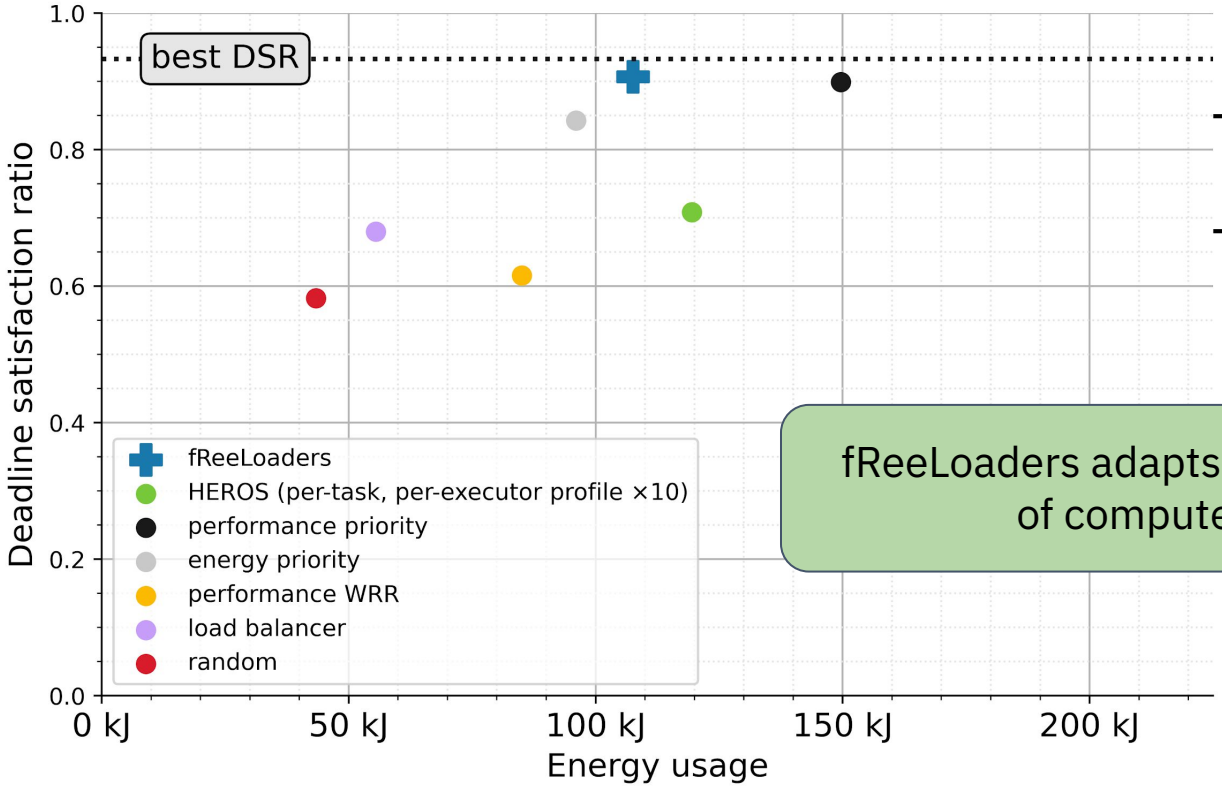


- vs. HEROS
- energy vs. DSR
- 4,000 tasks

fReeLoaders learns experientially



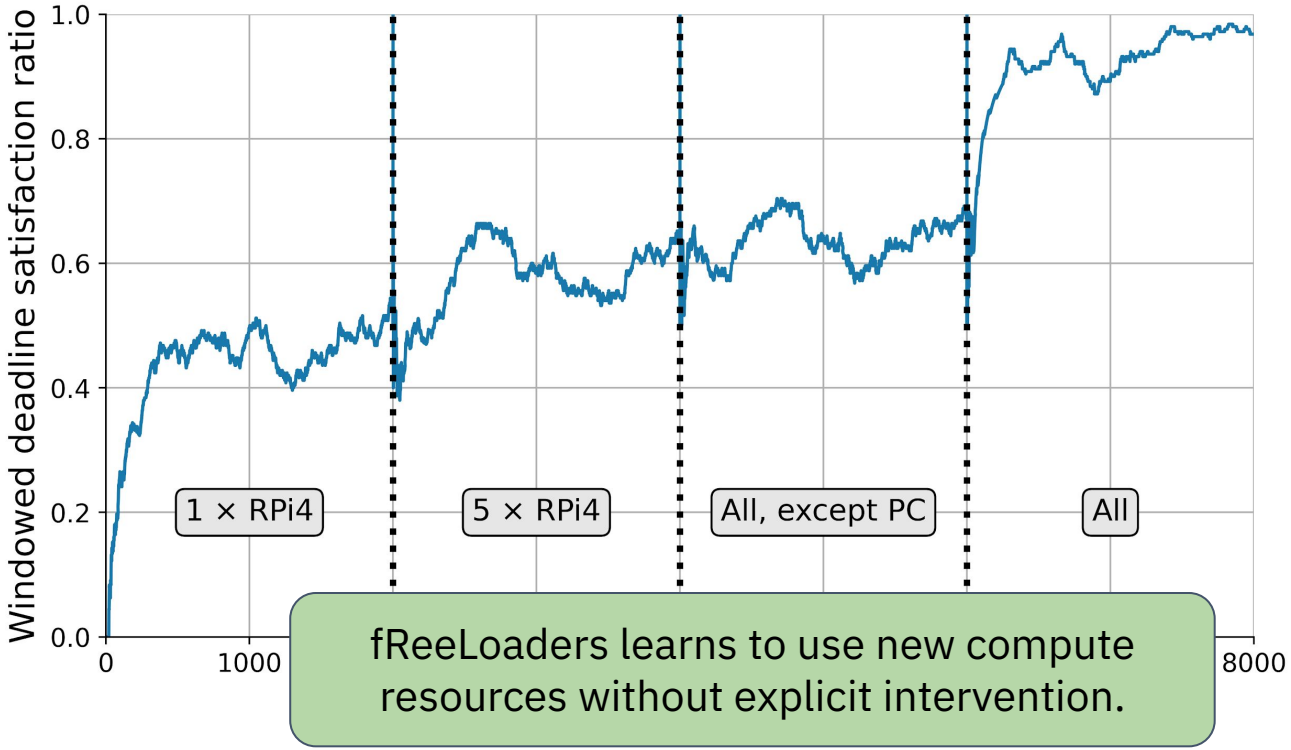
fReeLoaders adapts to non-dedicated compute



- Executors with various load patterns.
- All executors have periods of poor performance.

fReeLoaders adapts to the changing load of compute resources.

fReeLoaders adapts to adding executors

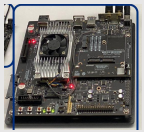


fReeLoaders learns to use new compute resources without explicit intervention.

available executors

RPi 4  x 5

RPi 3  x 2



TX2 Nano PC

Conclusions

- fReeLoaders **removes *a priori* profiling**, is effective at scheduling.
  - Don’t profile, **learn experientially**.
- On-the-fly adaptation enables success:
  - Works for heterogeneous compute resources.
  - Sensitive to computing environment dynamics (load, new hosts).
- An ecosystem to serve advancing real-time IoT applications.

Paper

Code

Additional Slides

Task generation

offloaders | tasks | controller | executors

workout data analysis

Task

- task type ID
- deadline
- data

fReeLoaders scheduler

Executor selection

offloaders | tasks | controller | executors

The scheduler requires only the load value from the executors, avoiding deep integration on executors.

task

- task type ID
- task deadline
- executor loads

→ fReeLoaders RL scheduler → executor ID

Execution feedback

offloaders | tasks | controller | executors

This feedback allows the RL scheduler to adapt to its dynamic environment.

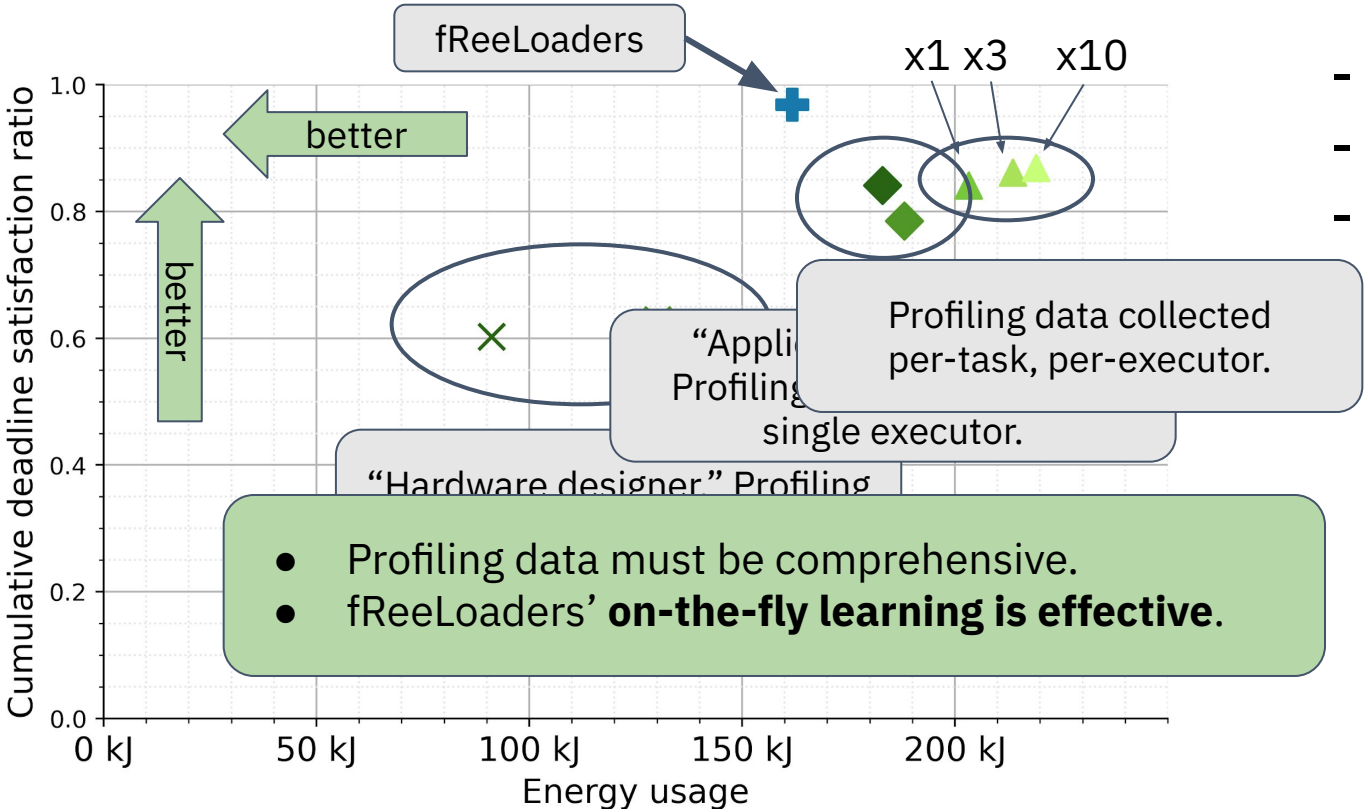
result

fReeLoaders RL scheduler

Reward
deadline met: [0, 1]
deadline miss: -1

- task type ID
- execution time
- energy used

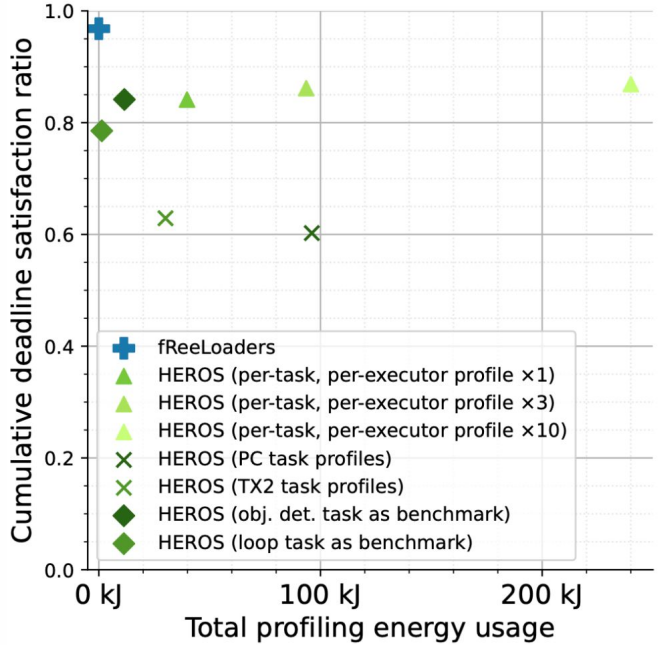
A performant alternative to profiling



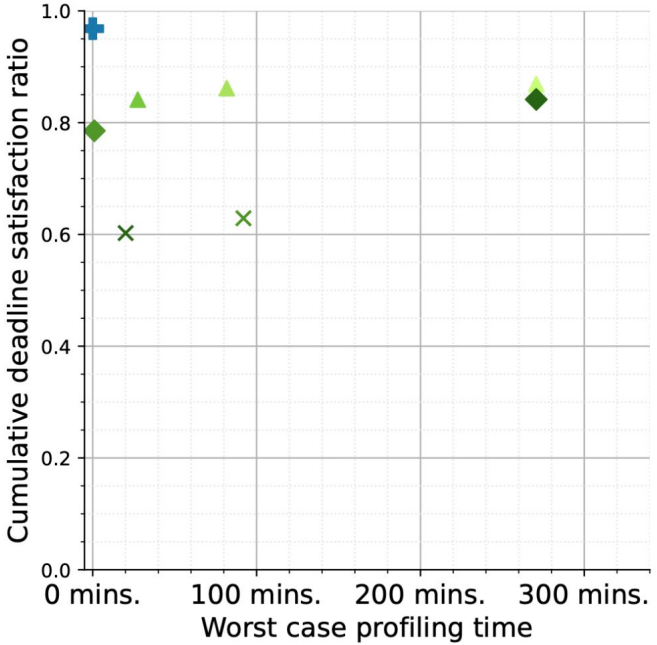
- vs. HEROS
- energy vs. DSR
- 4,000 tasks

fReeLoaders removes profiling overheads

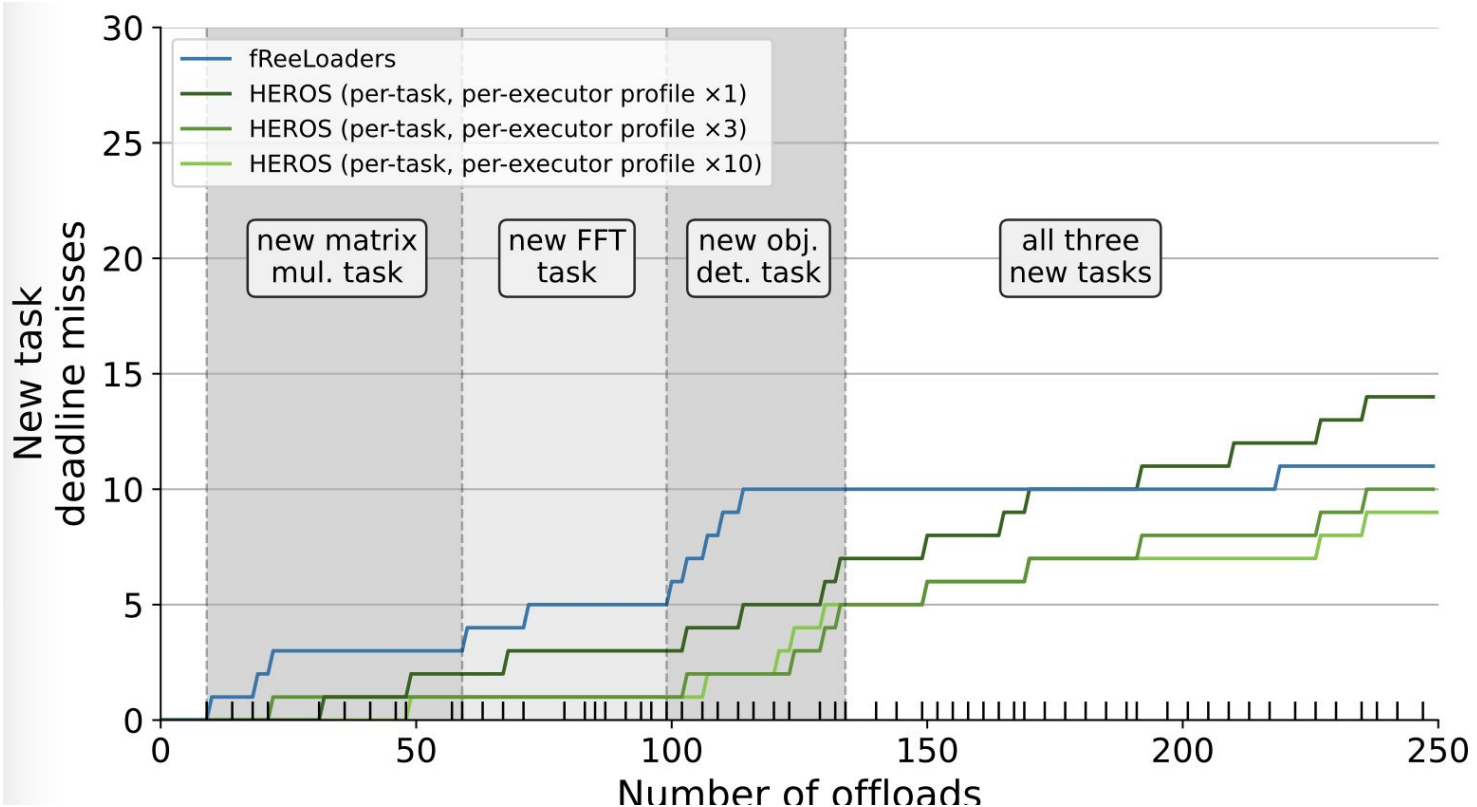
profiling energy overhead



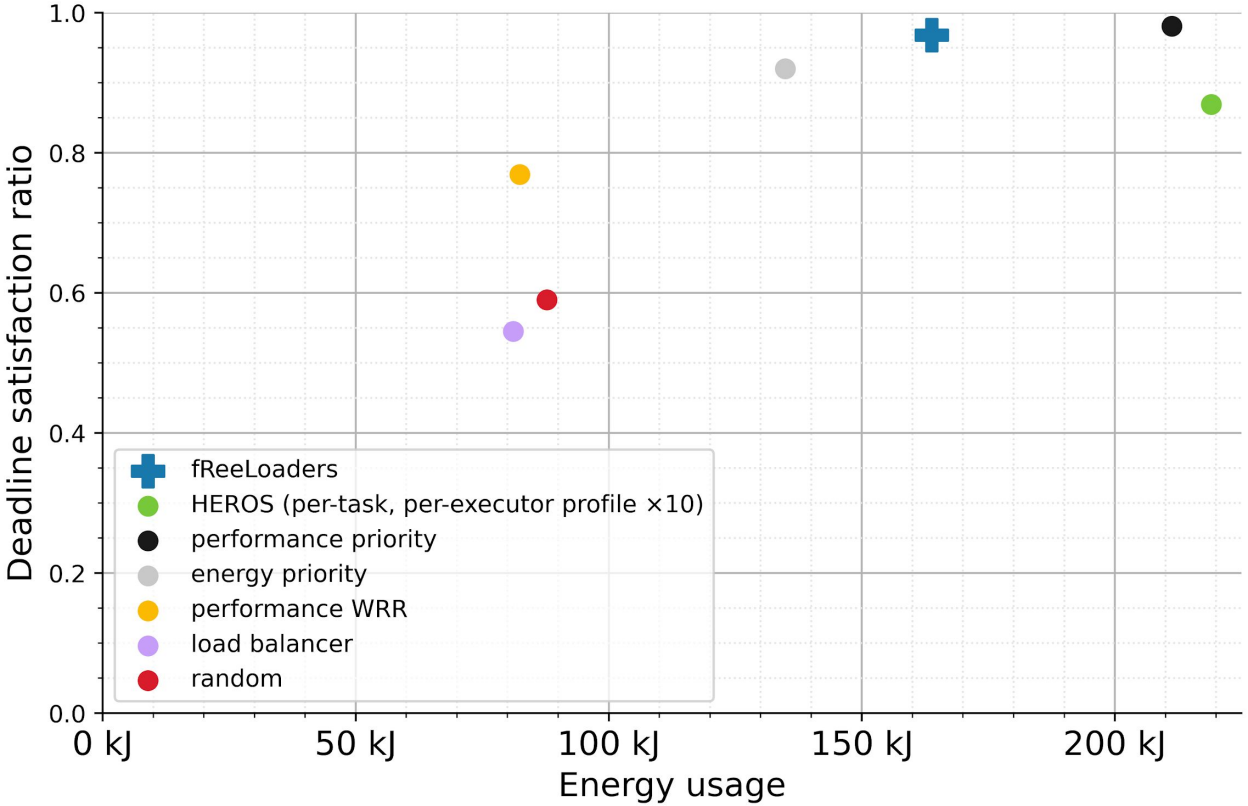
profiling time overhead



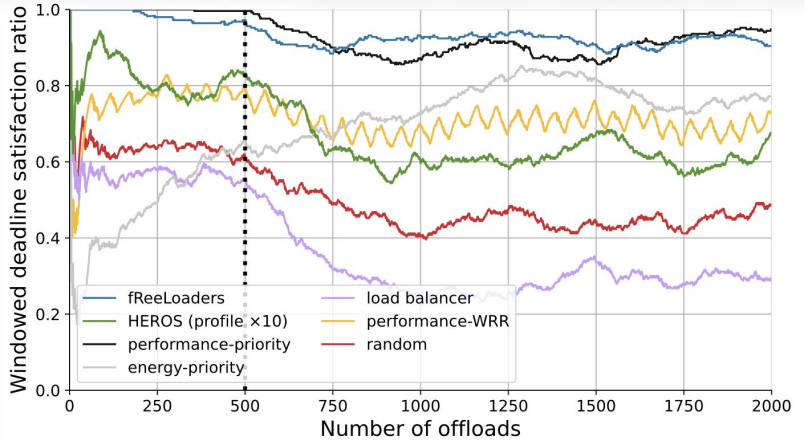
fReeLoaders learns new tasks



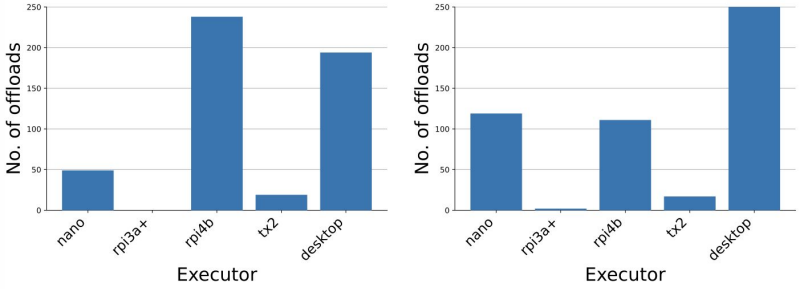
fReeLoaders vs. common approaches



fReeLoaders reacts to tightened deadlines



(a) Deadline satisfaction ratio.



(b) fReeLoaders task distribution by executor type for the first 500 offloads (left) and the last 500 offloads (right).

Case study: audio classification offloading