Energy Efficiency in Diverse Hardware and Software

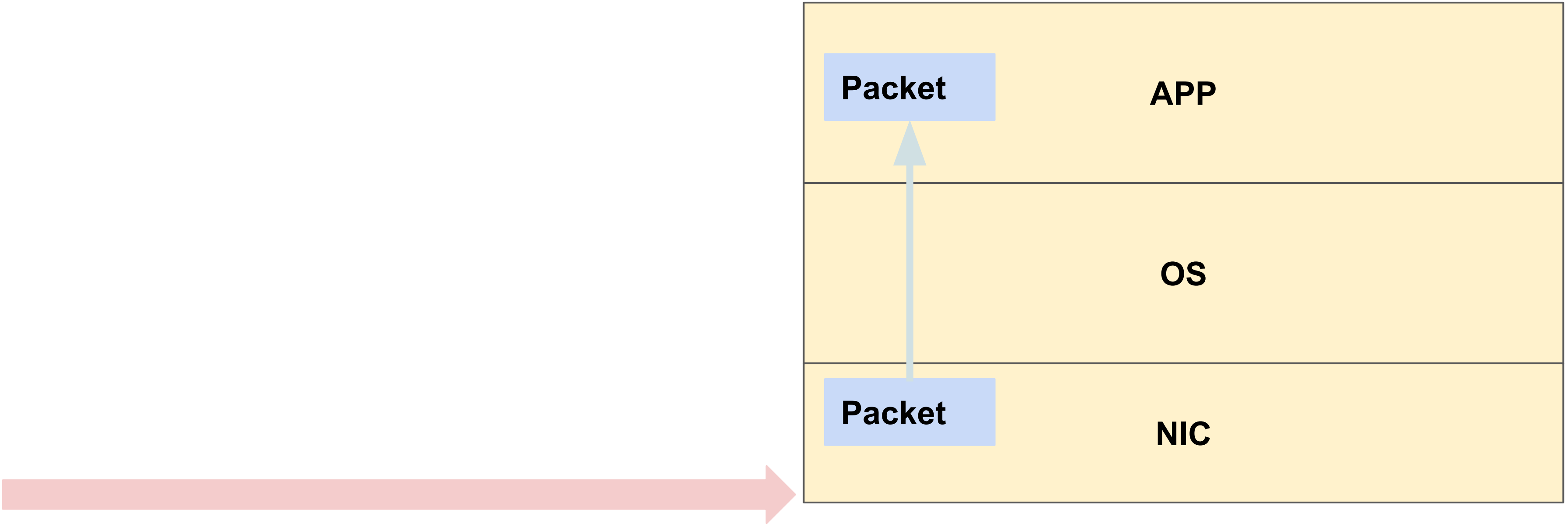
Han Dong, Sanjay Arora, Jonathan Appavoo
handong@bu.edu

BOSTON UNIVERSITY

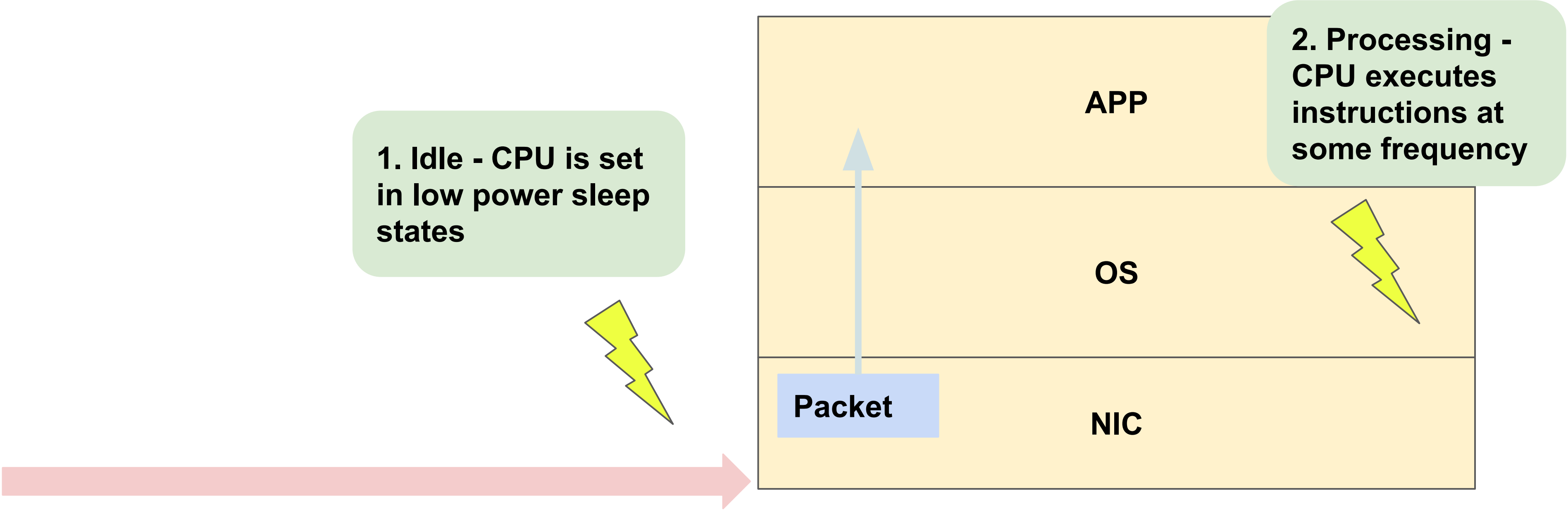
About me

- Defended my PhD thesis in 2023 at Boston University
- Currently a postdoc with Boston University/Red Hat
- Research interests:
  - Operating system design, implementation and optimization (systems hacking)
  - Measurement and analysis of systems
  - Applying ML techniques to improve computer systems

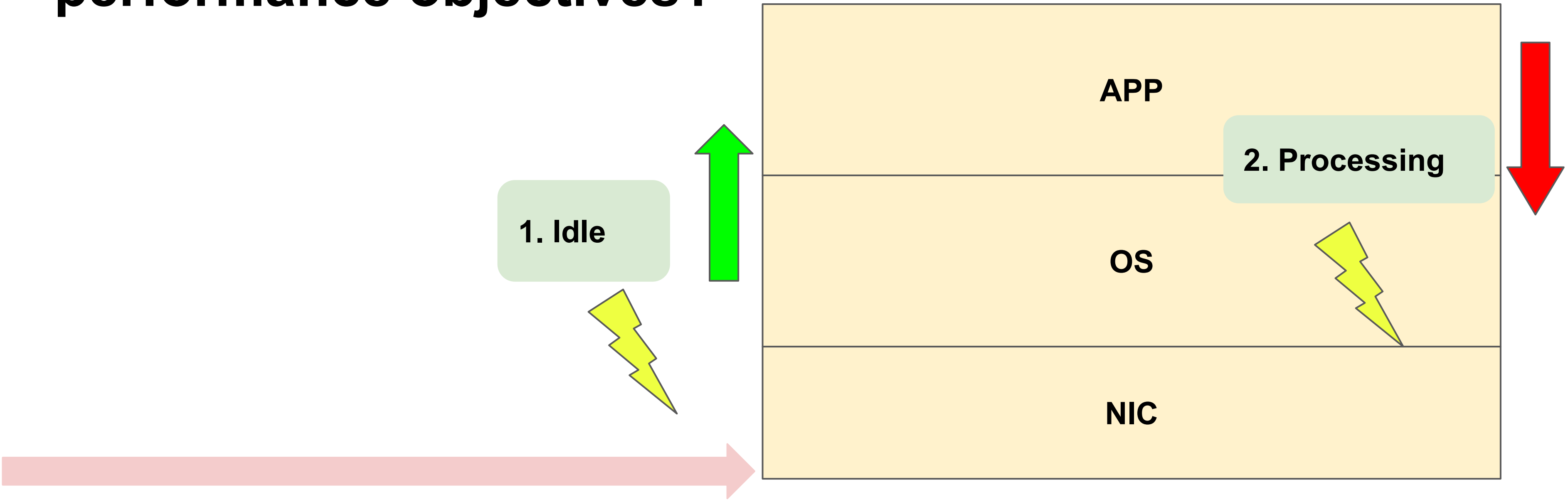
A simple network processing example



Where does energy consumption happen?

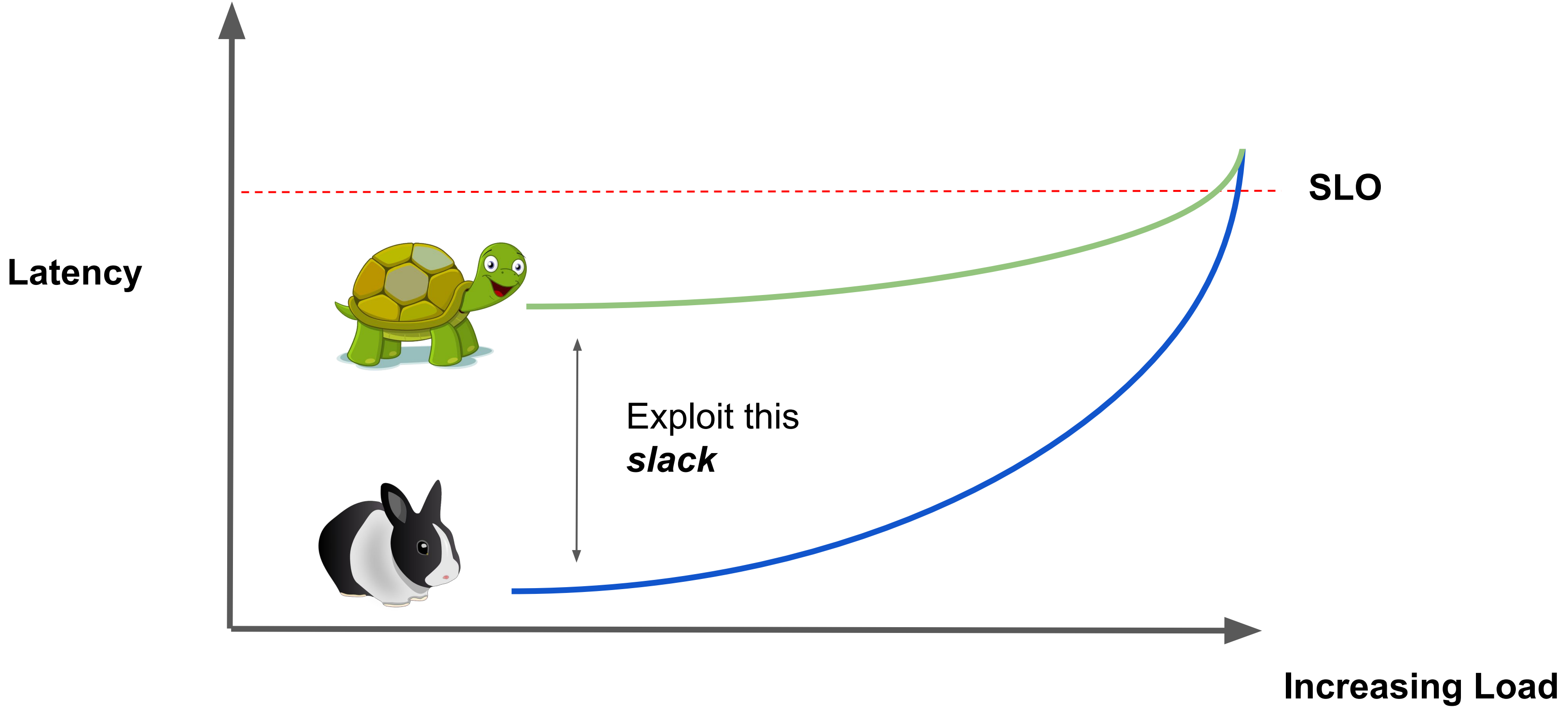


Can we reduce energy consumption while maintaining performance objectives?



Idle longer, slow CPU processing?

Web services with service-level objectives (SLO) i.e. 99% tail latency < 1 ms

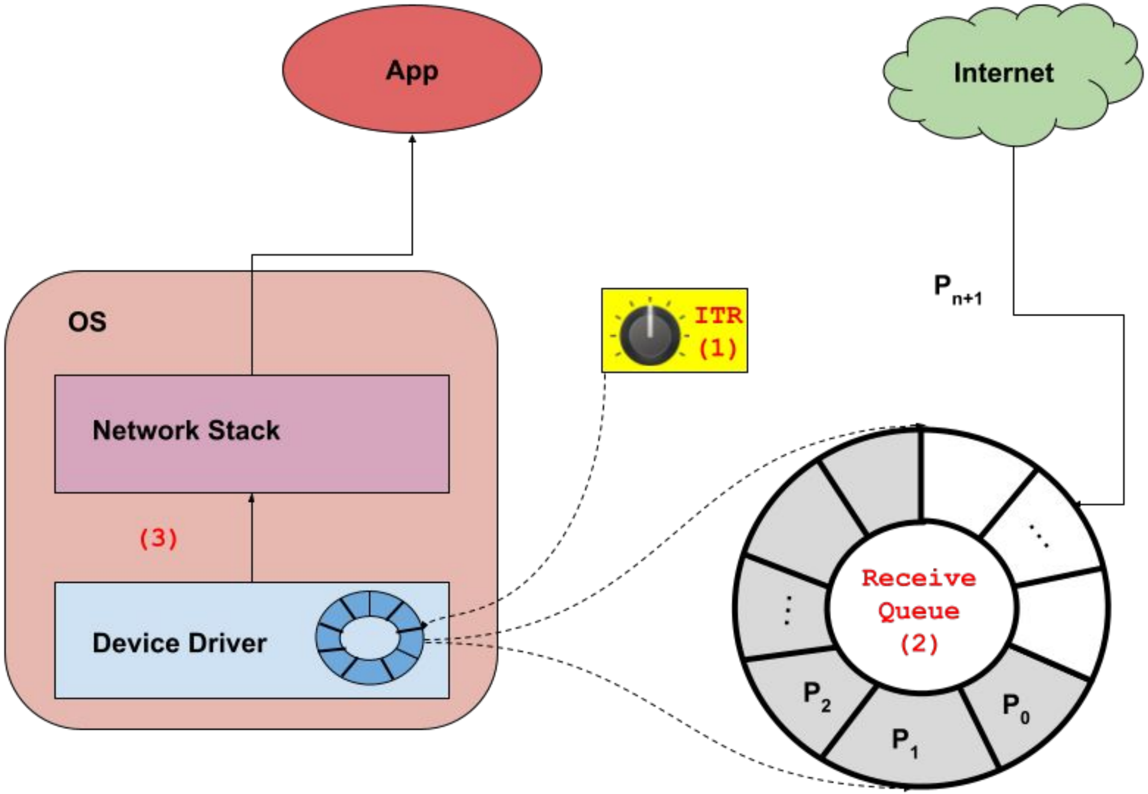


Our work: the novel tuning of two hardware knobs: 1) **interrupts coalescing** and 2) **processor speeds** to reduce energy while maintaining SLO across diverse sets hardware/software

Interrupt Coalescing (ITR)

Control packet batching behavior via Linux *ethtool*:

1. Set ITR value
2. Incoming packets are buffered on receive queue until ITR value has been reached
3. Network device asserts interrupt for packet processing
4. Set **dynamically** by NIC device driver on per-interrupt basis



Processor speed: Dynamic Voltage Frequency Scaling (DVFS)

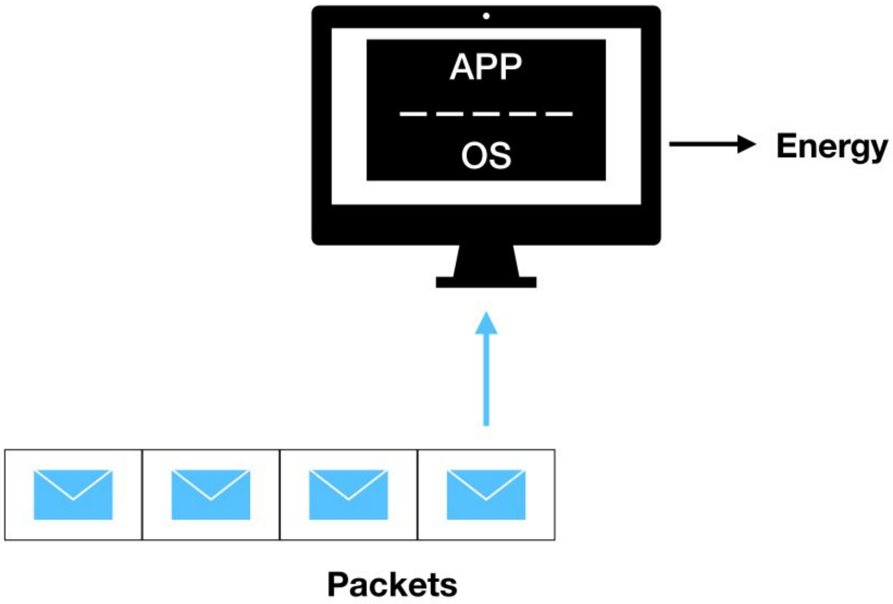
- $P = C \times V^2 \times f$
  - P = dynamic power
  - C = switching capacitance of logic
  - V = operational voltage
  - f = operational frequency

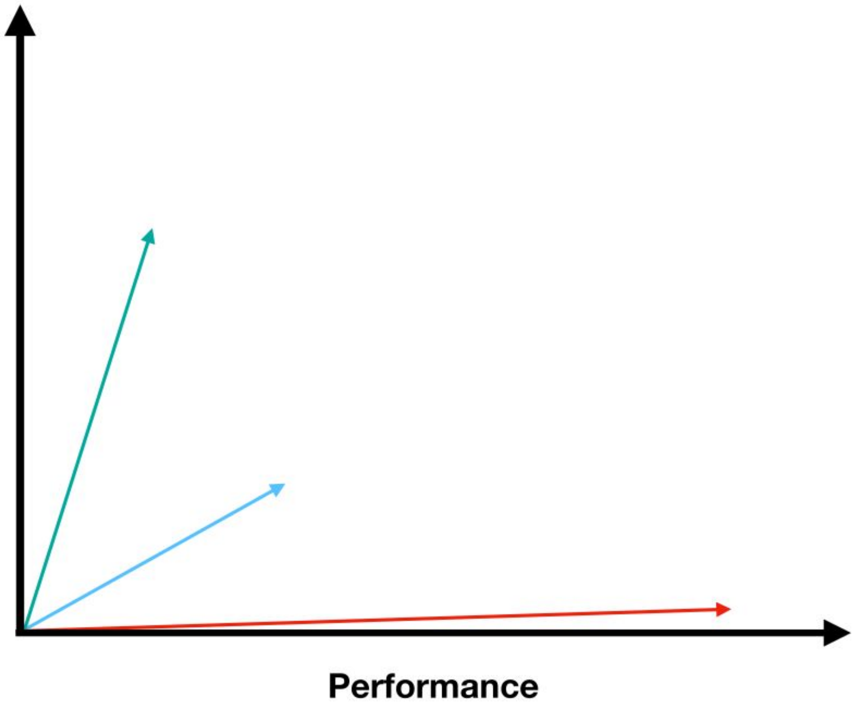
1. Set **dynamically** by Linux policy governors:
   a. Our work explores static frequencies on per-workload basis

1. Performance and Energy Study

Defining the measurement problem

- Massachusetts Open Cloud (MOC)
- Intel(R) Xeon(R) CPU E5-2690 @ 2.90GHz
- Intel 82599ES 10-Gigabit SFI/SFP+ NIC



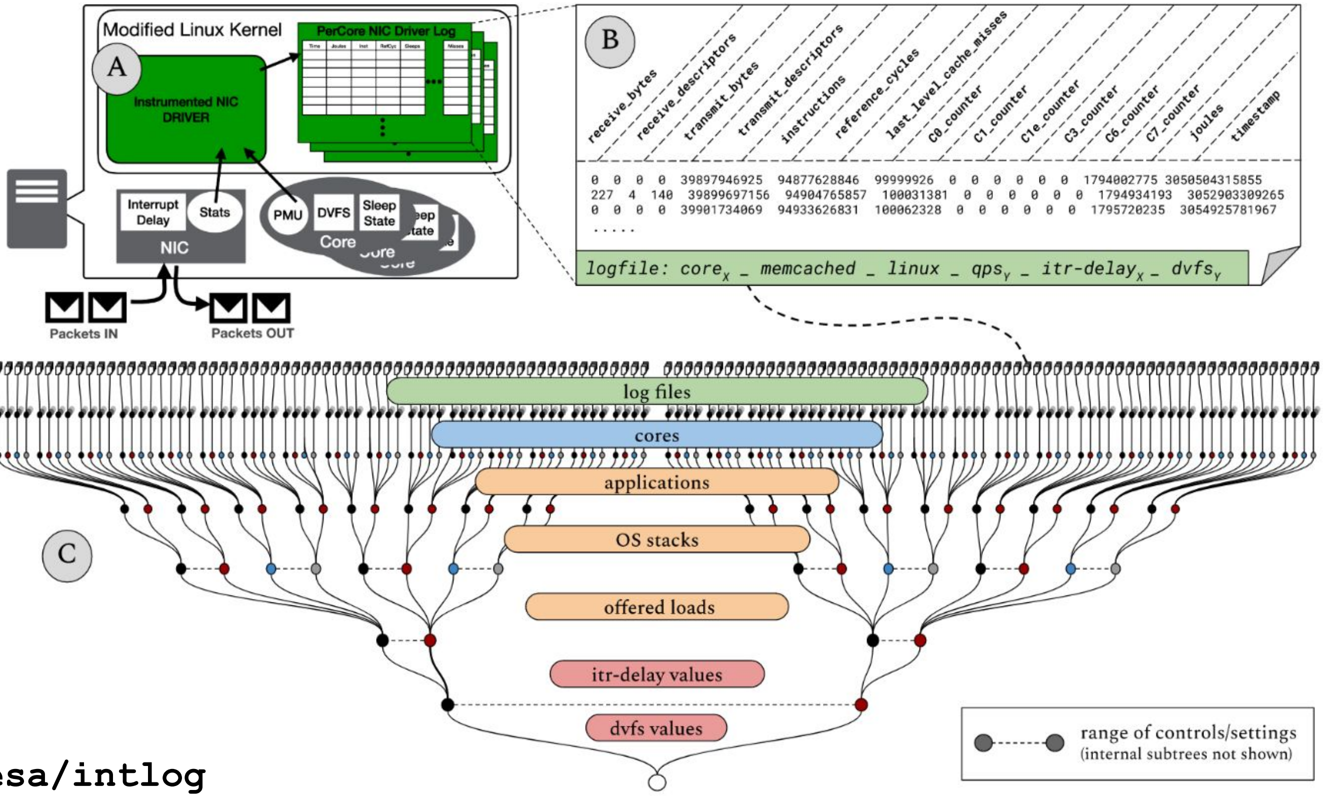


Data Collection Framework for Systems

A) Instrumented on network interrupt path

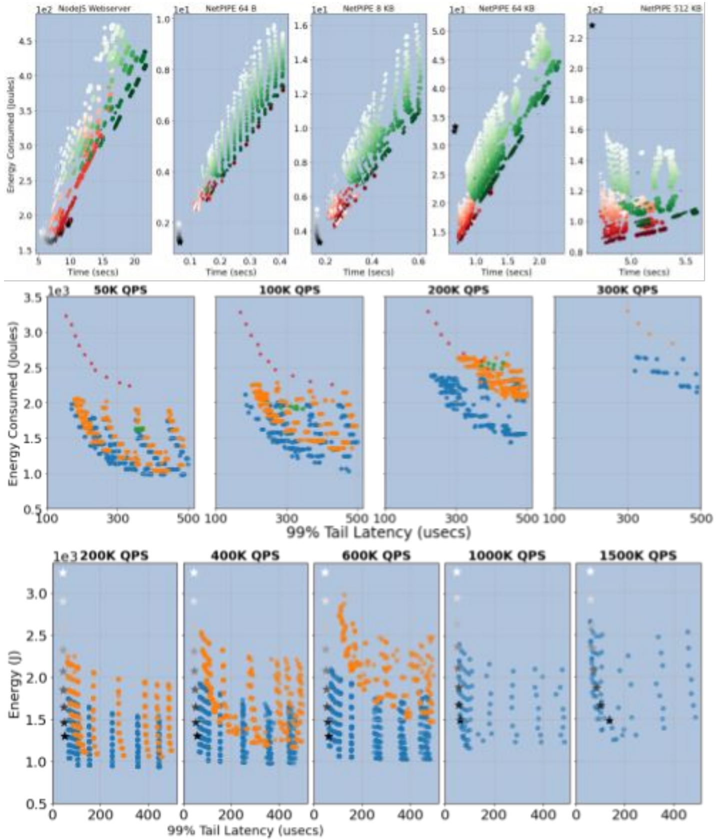
B) Per-core `/procfs` entry

C) Agnostic across applications/OSes



https://github.com/sesa/intlog

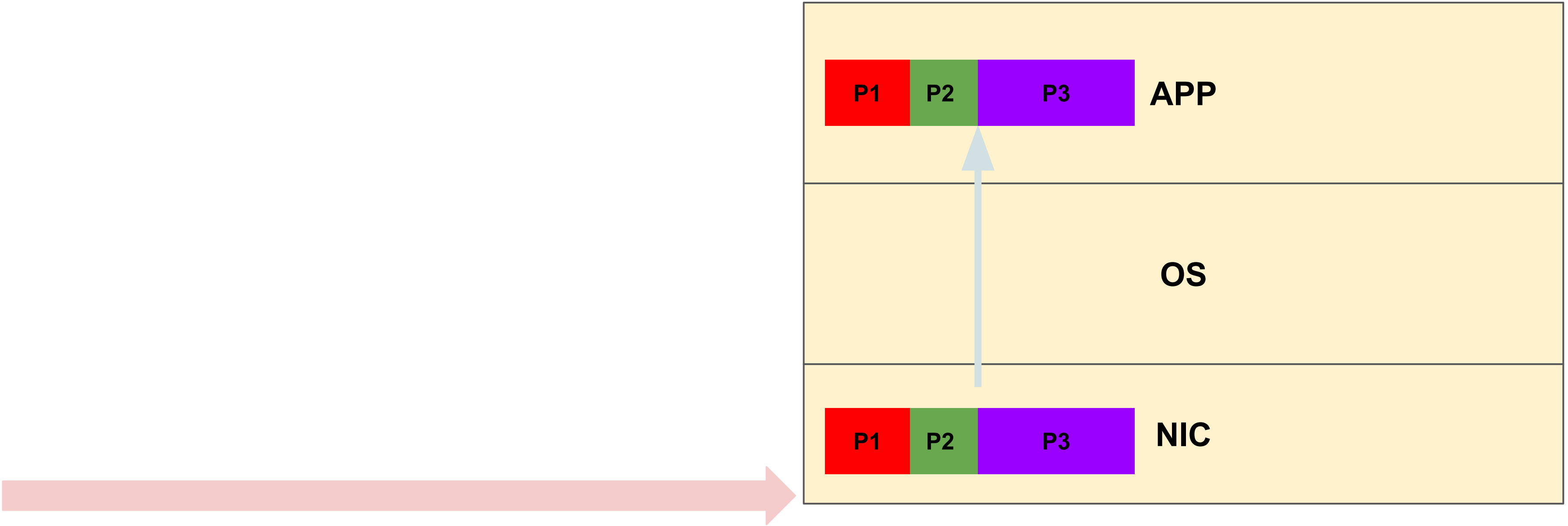
Data Collection Results



- Explored 340 unique ITR, DVFS combinations
- Repeated up to 10 times for experimental stability
- Collected over 5 TB of systems log data

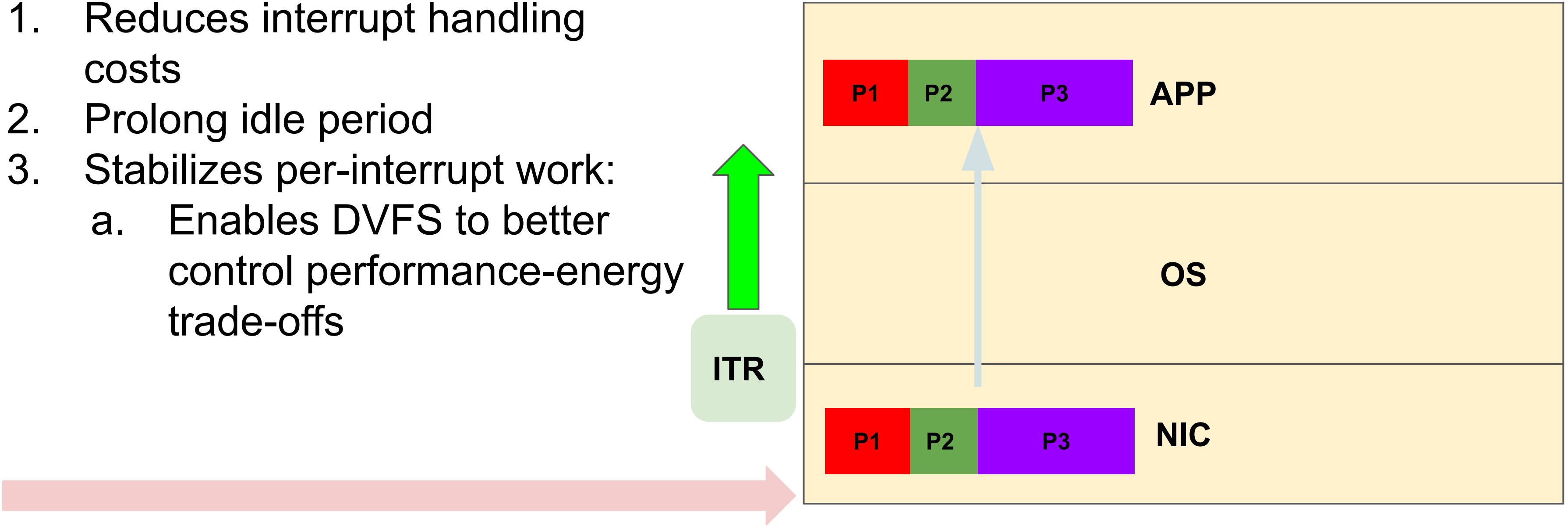
Finding: Linux can save up to <u>76% energy</u> while maintaining SLO of Memcached (https://arxiv.org/pdf/2112.07010.pdf).

Insight



Insight

1. Reduces interrupt handling costs
2. Prolong idle period
3. Stabilizes per-interrupt work:
   a. Enables DVFS to better control performance-energy trade-offs



2. Mathematical Modeling and Fitting

Motivation

1. Study reveals **characteristic common and stable structure** in response to changes to ITR, DVFS

2. Implication is that one can capture performance and energy profiles using an analytical model in a formal way:
   a. Suggests that **controlled learning** of policies for OSes can be made **feasible**

Mathematical Modeling Result

$$t_{\text{interrupt}} = ITR^{\gamma * DVFS + \delta}$$

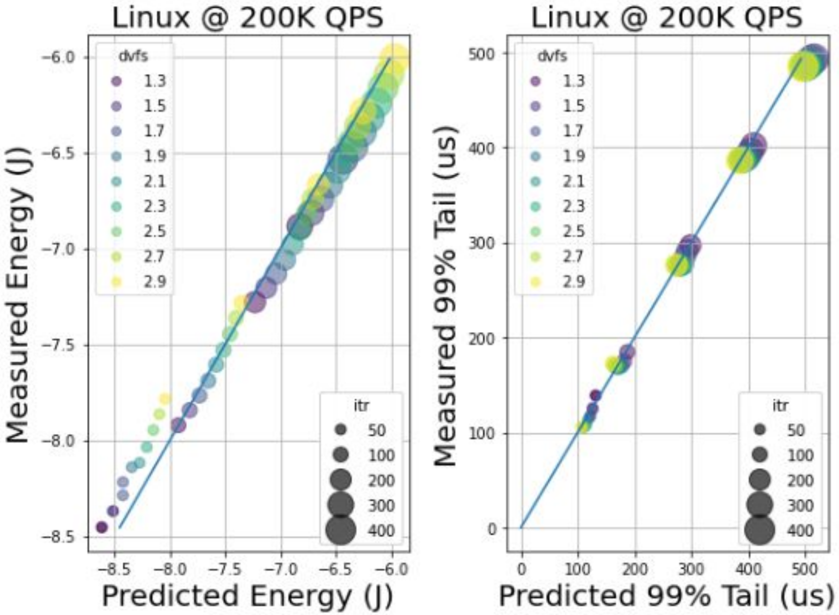
$$\triangle t = \frac{Z}{DVFS^{1+\alpha}} + (\phi * ITR)$$

$$\triangle J = \gamma * (\phi * ITR) * DVFS^{\beta}$$

$$\triangle J = 2 * (CF_a * t_{interrupt} + CF_b * t_{transmit} + CF_c * t_{work})$$

$$\Delta t = 2 * (\frac{ITR^{\gamma * DVFS + \delta}}{10^6} + t_{transmit} + \frac{Z}{DVFS^{1+\alpha}})$$

$$\Delta J = CF_a * (\frac{ITR^{\gamma * DVFS + \delta}}{10^6}) + CF_b * (t_{transmit}) + CF_c * (\frac{Z}{DVFS^{1+\alpha}})$$

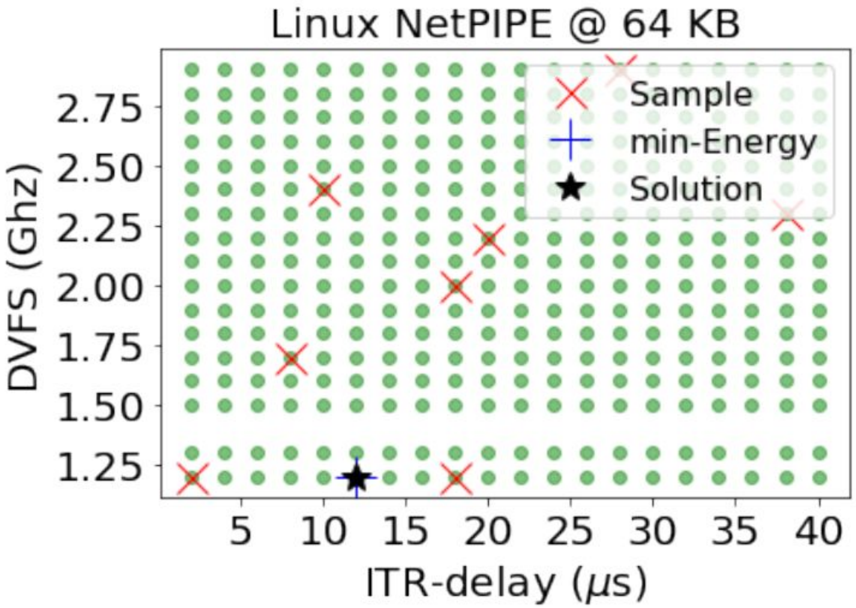


Details can be found in thesis*

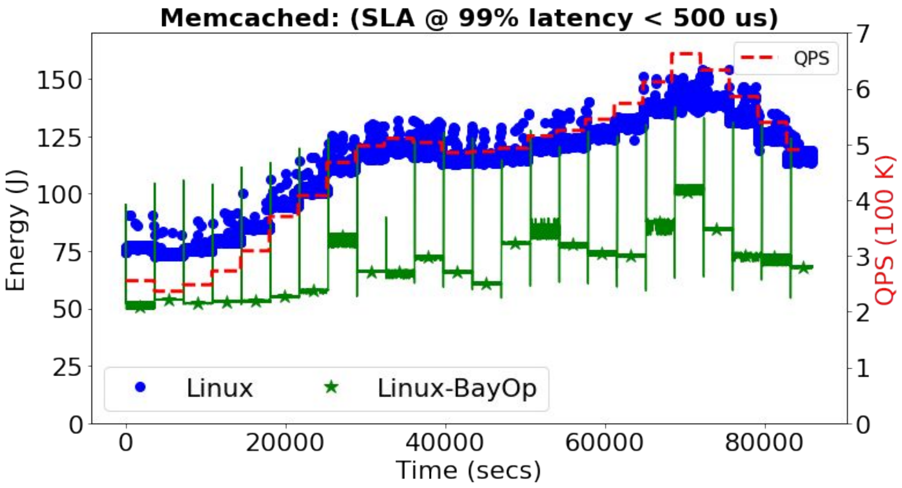
3. Applied ML for ITR, DVFS

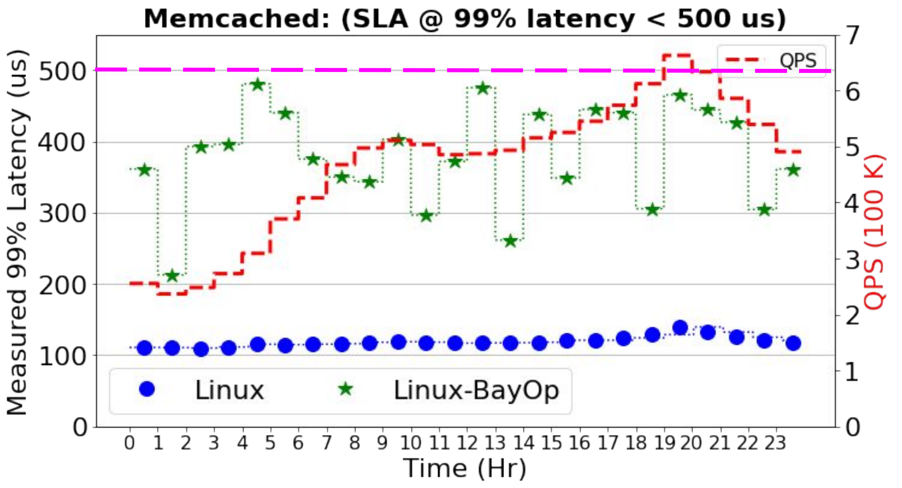
Building Block: Sample Efficient Machine Learning

Reduces the number of samples to find some optimal configuration:



Applying ML to Twitter *cache-trace* Dataset





Details: ***https://research.redhat.com/quarterly/***

4. Generality of ITR, DVFS tuning

Diverse hardware and software

Hardware:

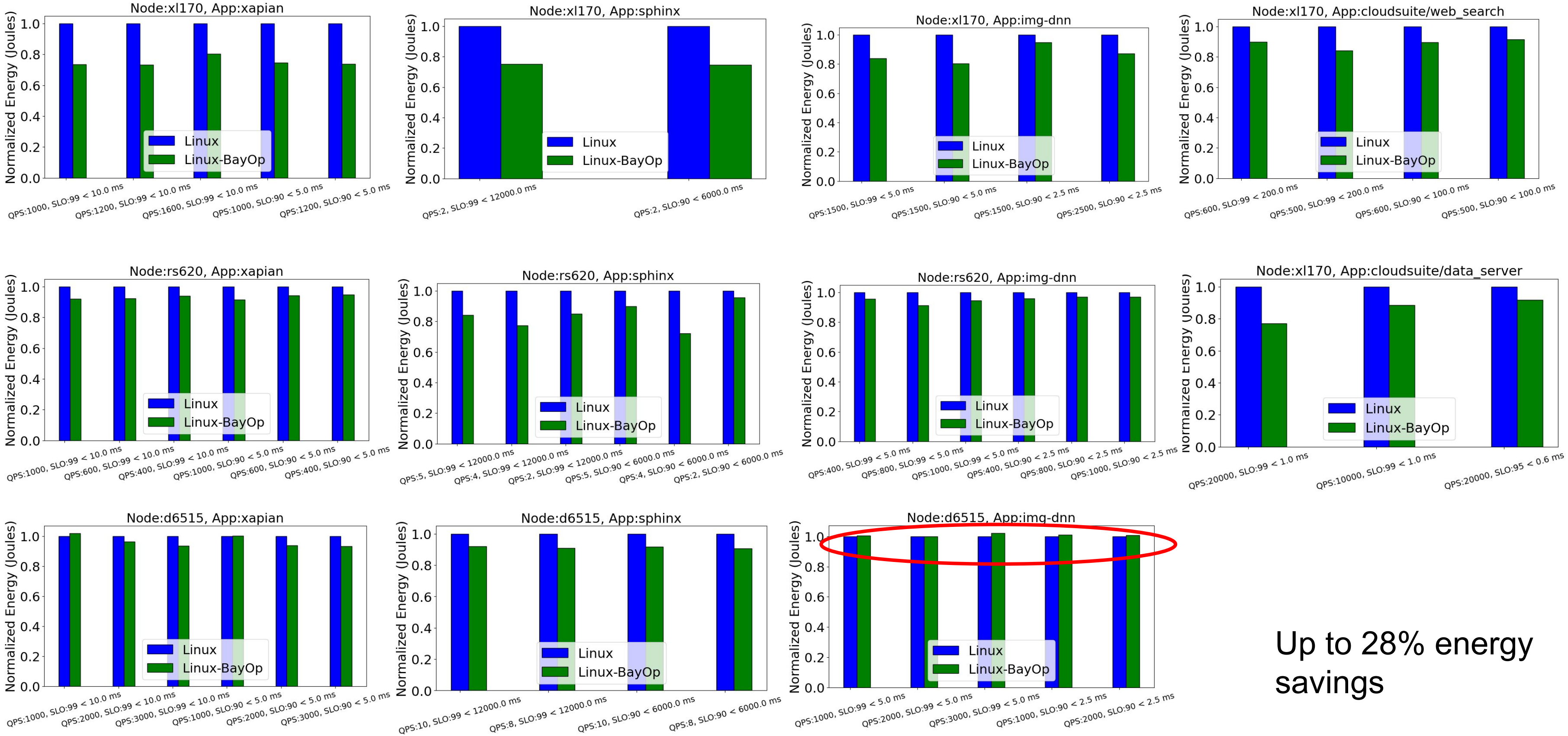
1. **xl170**: Intel(R) Xeon(R) CPU E5-2640, Mellanox 25 GB NIC
2. **rs620**: Intel(R) Xeon(R) CPU E5-2660, Solarflare 10GB NIC
3. **d6515**: AMD EPYC 7452, Mellanox 40 GB NIC
4. Q80-30 Ampere ARM Altra processor, Mellanox 25 GB NIC (**in progress**)

Software:

a. **Tailbench** (http://tailbench.csail.mit.edu/)
   i. **Img-dnn**: handwriting recognition application based on OpenCV
   ii. **Xapian**: Open-source search engine
   iii. **Sphinx**: Speech recognition system

b. **Cloudsuite** (https://github.com/parsa-epfl/cloudsuite)
   i. **Data-serving**: Cassandra NoSQL database with Yahoo! Cloud Serving Benchmark (YCSB)
   ii. **Web-search**: Apache Solr search engine
   iii. **Web-serving**: MariaDB+Memcached+PHP backend

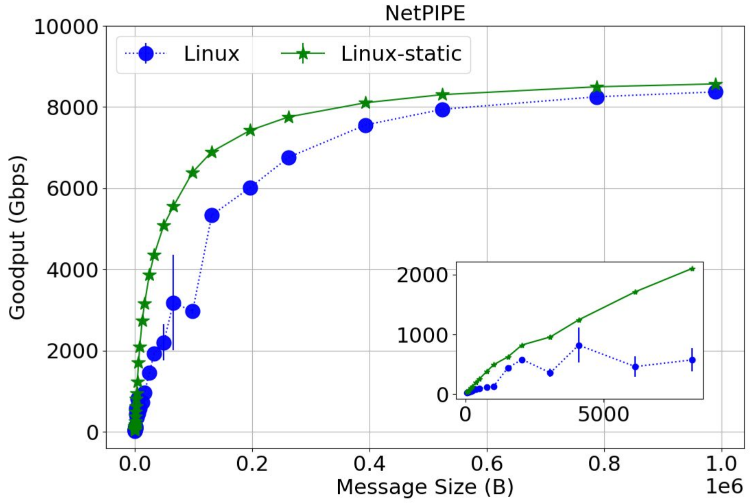
Results (in progress)

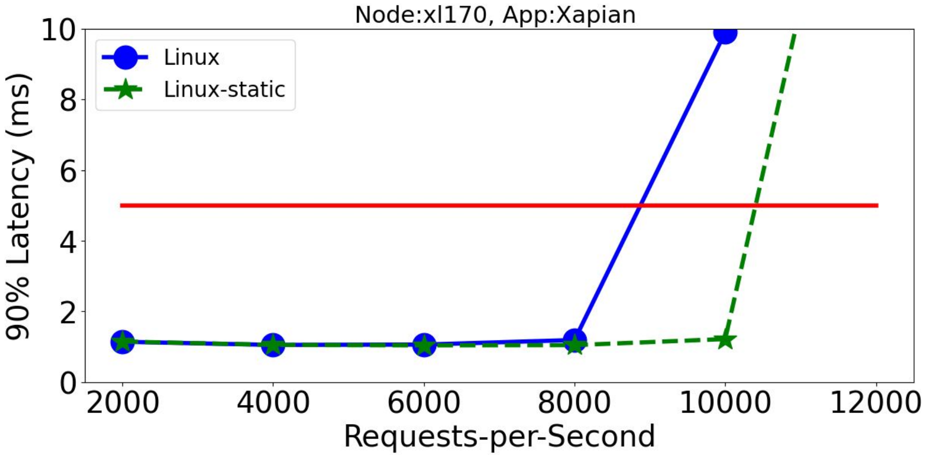


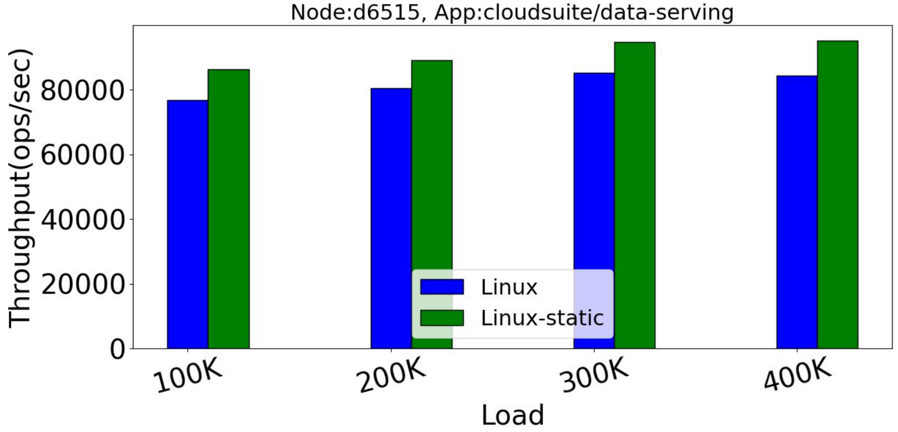
Up to 28% energy savings

Quick note about performance

- **Tuning ITR while DVFS at maximum performance can yield performance improvements up to 20%:**
    - **Requires more study**



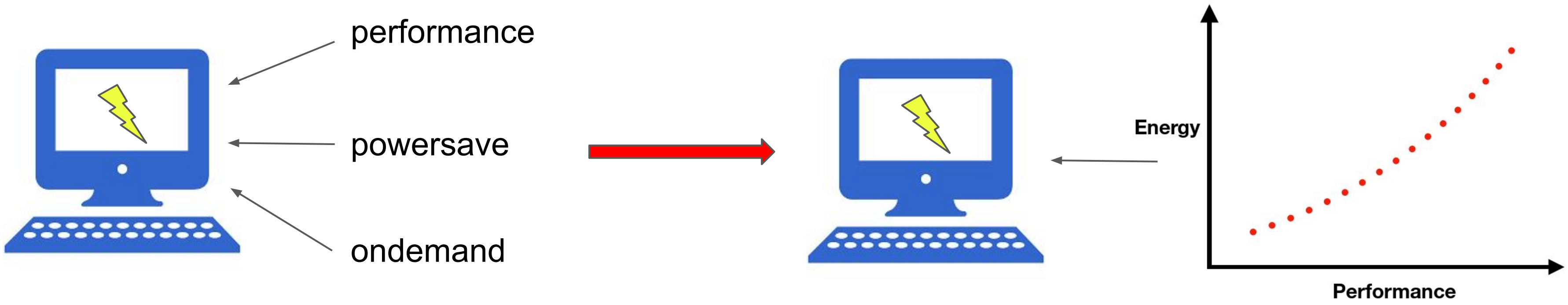




Next Steps

Short Term

1. Replace existing dynamic ITR algorithms with more robust heuristics:
    a. Widen scope of system metrics to consider
    b. Profile a wide range of applications to yield insight towards new policies
2. Create new energy efficient system policies:

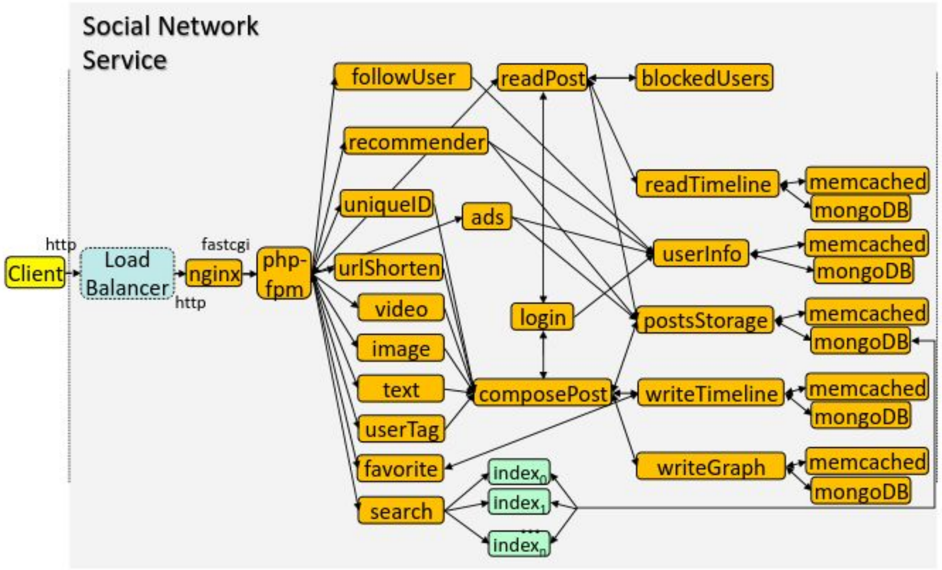


Medium Term

1. Add C-states tuning in conjunction with ITR
2. Multiplexing different workloads on same machine with ITR, DVFS tuning:
    a. Interrupt steering
    b. Per-core DVFS
    c. Per-core ITR
    d. Multiple SLOs

Long Term

1. Distributed system for multi-node tuning
2. Other hardware: SSD, GPUs, etc



https://www.csl.cornell.edu/~delimitrou/papers/2019.asplos.microservices.pdf