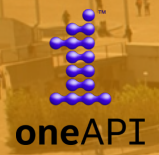
# Energy-Efficient Heterogenous Computing with SYNERGY




**oneAPI DevSummit for AI and HPC 2023**

December 6, 2023


Biagio Cosenza

Department of Computer Science, University of Salerno, Italy

Khronos SYCL Working Group Member

web www.cosenza.eu, email bcosenza@unisa.it

# Outline

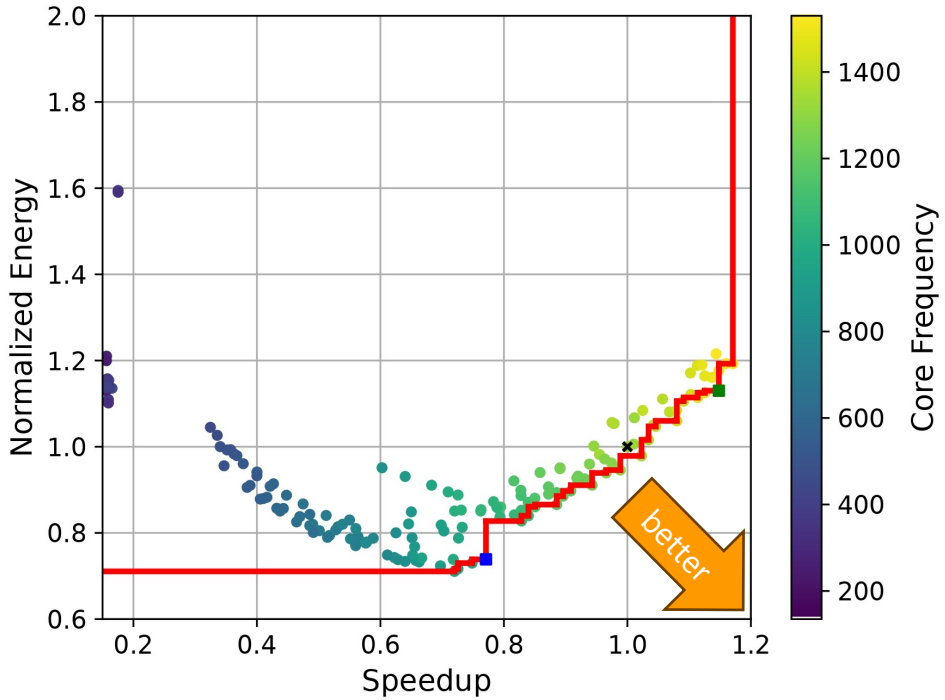
- Introduction
- Background
- SYnergy Approach Overview
  - API
  - Compilation
  - Runtime
- Conclusions

# Introduction

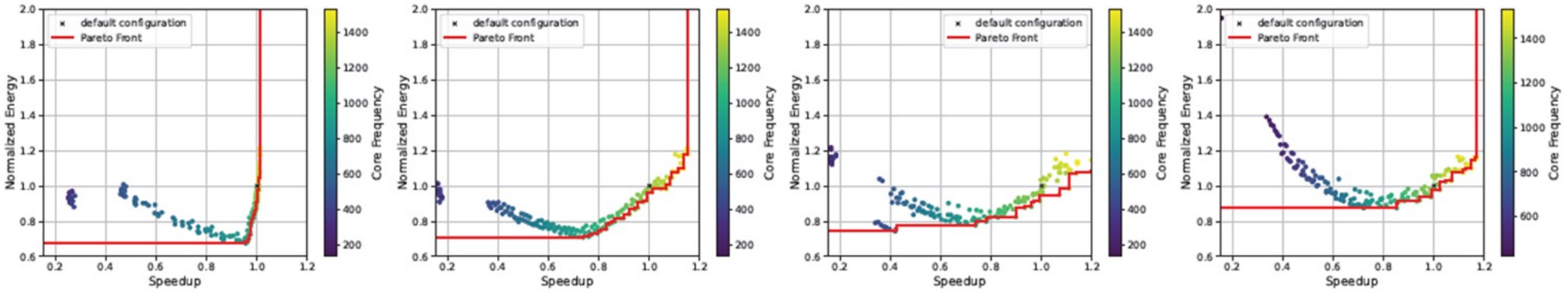
- Energy-efficient computing: one of the top ten research challenges in HPC
- Insights from the state of the art
  - Dynamic Voltage and Frequency Scaling (DVFS) aims to reduce power consumption by dynamically adjusting voltage and frequency
    - hardware vendor-specific power management library
    - e.g, LevelZero on Intel GPUs, NVML on NVIDIA GPUs, ROCm-SMI on AMD GPUs
    - no existing portable interface
  - Energy characterization depends on the kernel and on the hardware

# Background: Energy Optimization by DVFS

- Energy optimization is a multi-objective problem
  - There is no single frequency that achieves at best performance and energy consumption
  - Pareto-set of dominant solutions: we can exp can provide different trade-offs between speed
  - Default frequency is not optimal

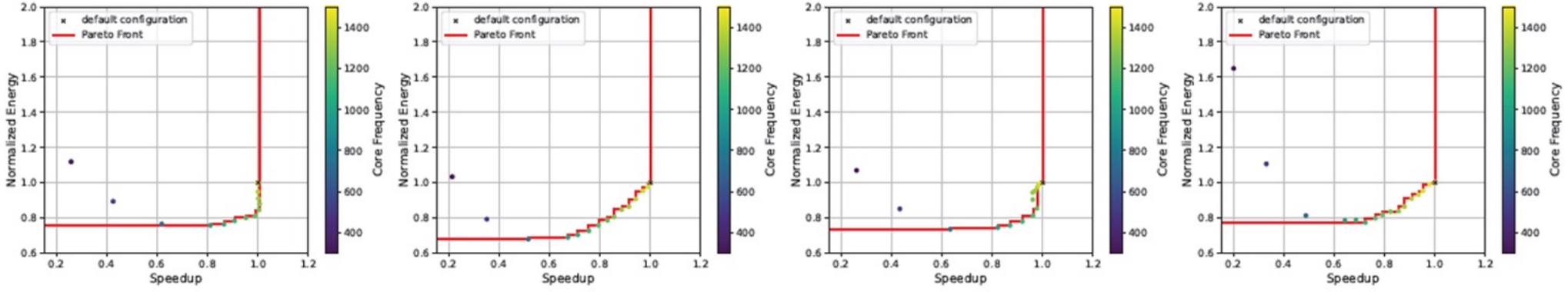


# Background: Energy Characterization on NVIDIA V100S and AMD MI100




(a) Matrix Multiplication (b) Sobel 3 (c) Mersenne Twister (d) FTLE

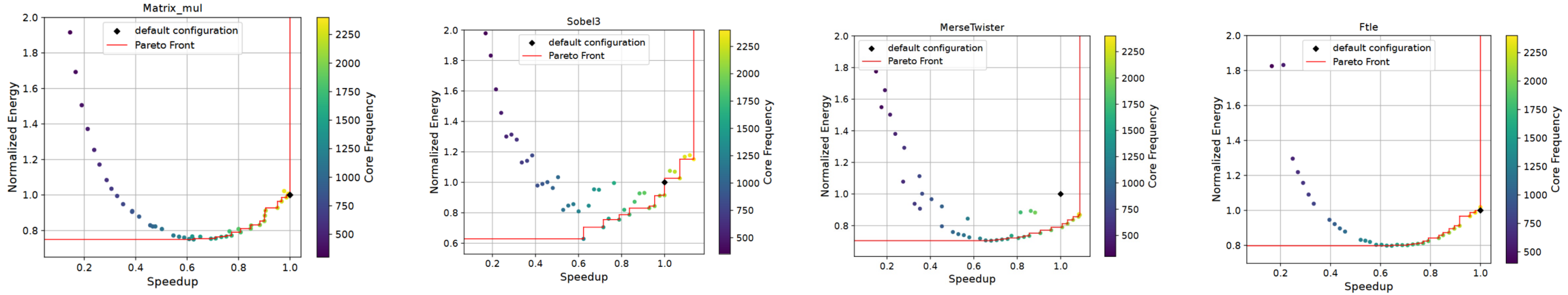
NVIDIA V100S



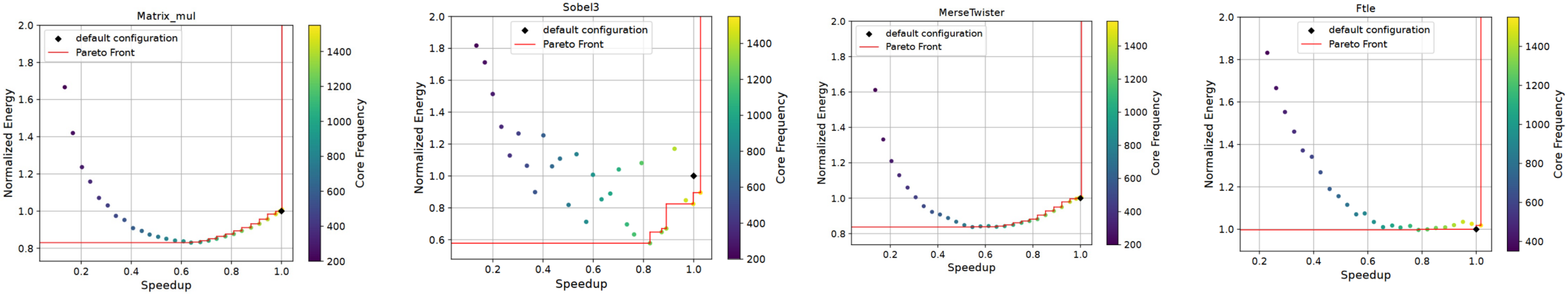

(a) Matrix Mutliplication (b) Sobel 3 (c) Mersenne Twister (d) FTLE

AMD MI100

# Background: Energy Characterization on Intel ARC A770 and Max 1100

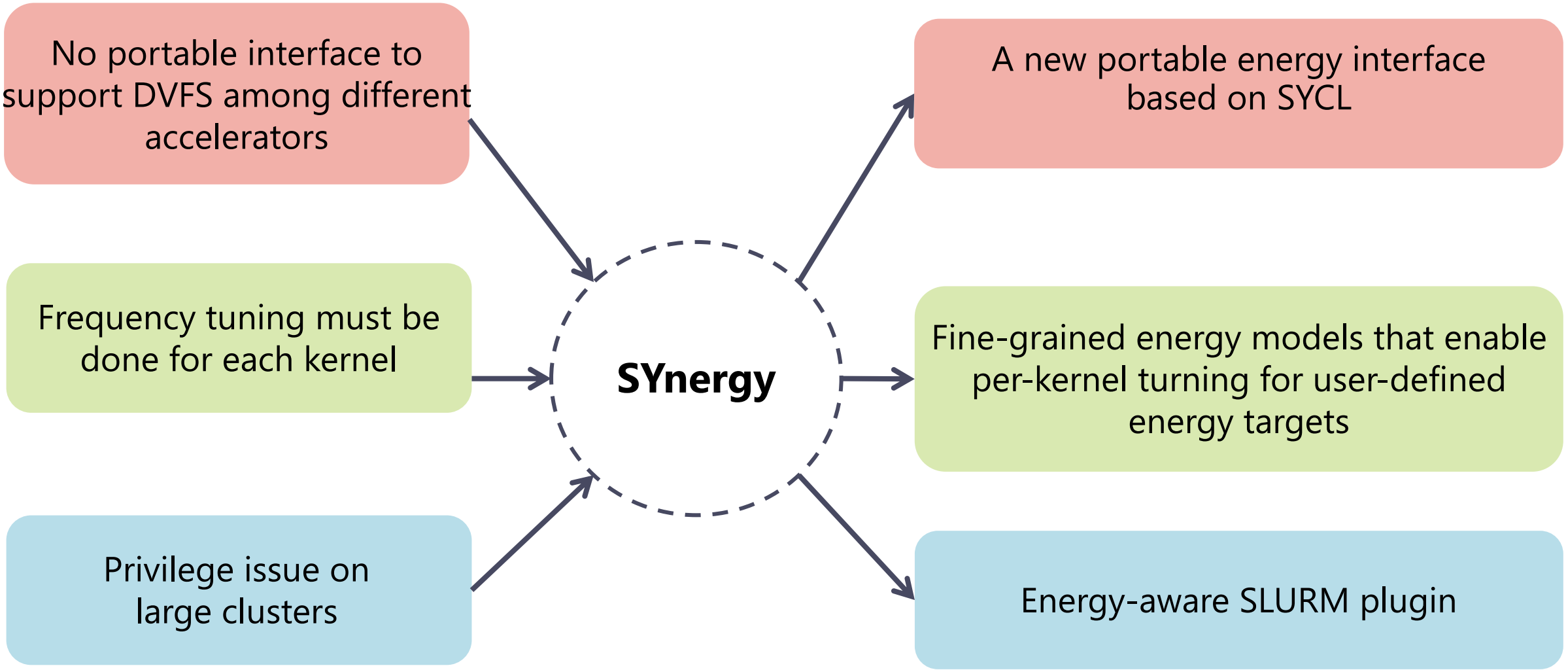



Intel ARC A770

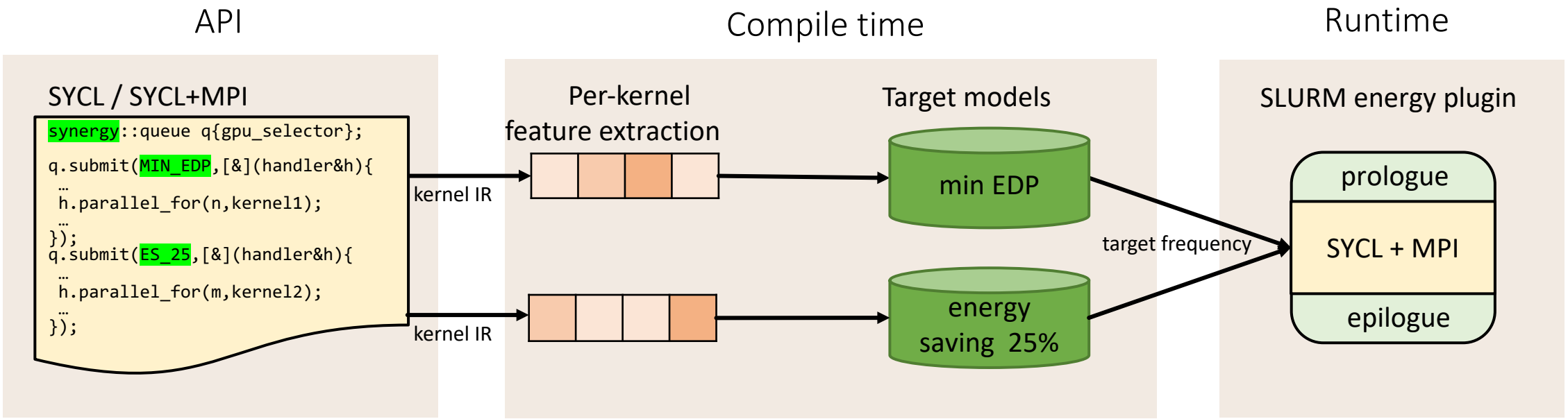



Intel Max Series 1100

# SYnergy: Challenges and Contributions



# SYnergy Overview




- SYnergy application: either SYCL or SYCL + MPI
  - using synergy::queue / specify energy target metric at queue or kernel level
- SYnergy program compilation
  - extracts a feature vector from each kernel / features fed machine learning energy models
- MPI+SYCL: SLURM energy plugin to support GPU frequency scaling on cluster

# Synergy API:

- Capability
  - energy profiling
  - frequency scaling
  - energy target setting

API

SYCL / SYCL+MPI

```
synergy::queue q{gpu_selector};
q.submit(MIN_EDP,[&](handler&h){
 …
 h.parallel_for(n,kernel1);
 …
});
q.submit(ES_25,[&](handler&h){
 …
 h.parallel_for(m,kernel2);
 …
});
```

# SYnergy API: Energy Profiling

- Energy semantics
  - Energy-aware queue
  - Fine-grained energy profiling
  - Coarse-grained energy profiling

```
synergy::queue q{gpu_selector_v};
buffer<float, 1> x_buf{x};
buffer<float, 1> y_buf{y};

event e = q.submit([&](handler& h) {
    accessor<float, 1, read> x_acc{x_buf, h};
    accessor<float, 1, read> y_acc{y_buf, h};
    float a{alpha};

    h.parallel_for(range<1>{n}, [=](id<1> id) {
    y_acc[id] = a * x_acc[id];
    });
});

double kernel_energy = q.kernel_energy_consumption(e);
double device_energy = q.device_energy_consumption();
```

SYCL code with SYNERGY queue

# SYnergy API: Frequency Scaling

- Frequency semantics
  - Energy-aware queue
  - Fine-grained frequency scaling
  - Coarse-grained frequency scaling

```
synergy::queue q1{1215, 210, gpu_selector_v};
synergy::queue q2{gpu_selector_v};
... // setup buffers

q1.submit([&](handler& h) {
   ... // setup accessors
   h.parallel_for(n,kernel1);
});

q2.submit(877,810, [&](handler& h) {
  ... // setup accessors
  h.parallel_for(m,kernel2);
});
```

SYCL code with SYNERGY queue

# SYnergy API: Energy Targets

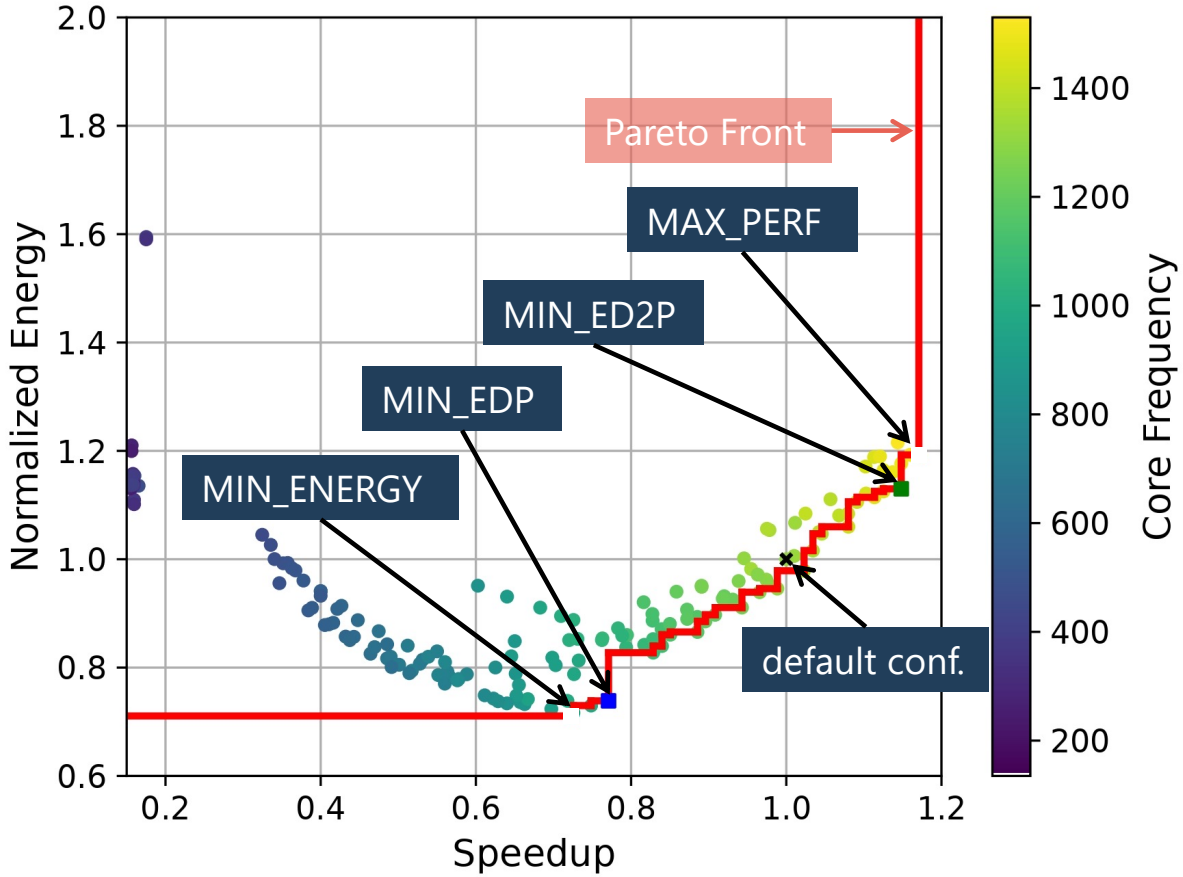
- Energy target semantics
  - Energy-aware queue
  - Per-kernel energy target, e.g., `MAX_PERF`, `MIN_ENERGY`, `MIN_EDP`, or `MIN_ED2P`

```
synergy::queue q{gpu_selector_v};
buffer<float, 1> x_buf{x};
buffer<float, 1> y_buf{y};

event e = q.submit(MIN_EDP,[&](handler& h) {
    accessor<float, 1, read> x_acc{x_buf, h};
    accessor<float, 1, read> y_acc{y_buf, h};
    float a{alpha};

    h.parallel_for(range<1>{n}, [=](id<1> id) {
    y_acc[id] = a * x_acc[id];
    });
});
```

SYCL code with SYNERGY queue

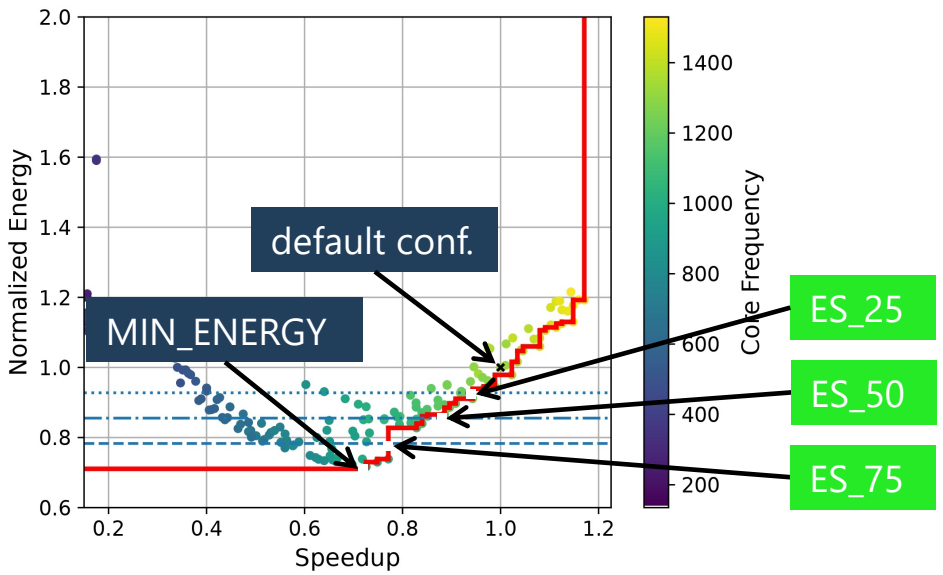
# SYnergy API: Traditional Energy Targets

- Scalar metrics
  - `MAX_PERF`, `MIN_ENERGY`, `MIN_EDP`, `MIN_ED2P`
- Difficulty to represent energy-performance tradeoff
- Interesting configurations can be found in the multi-objective distribution

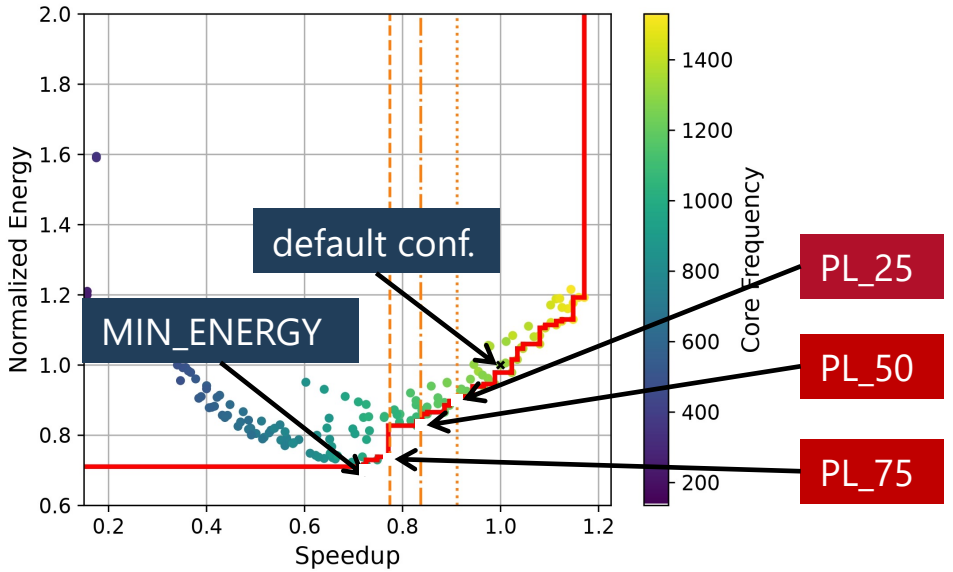



Energy targets of Black-Scholes benchmark

# SYnergy API: Novel Energy Targets




Energy-saving targets

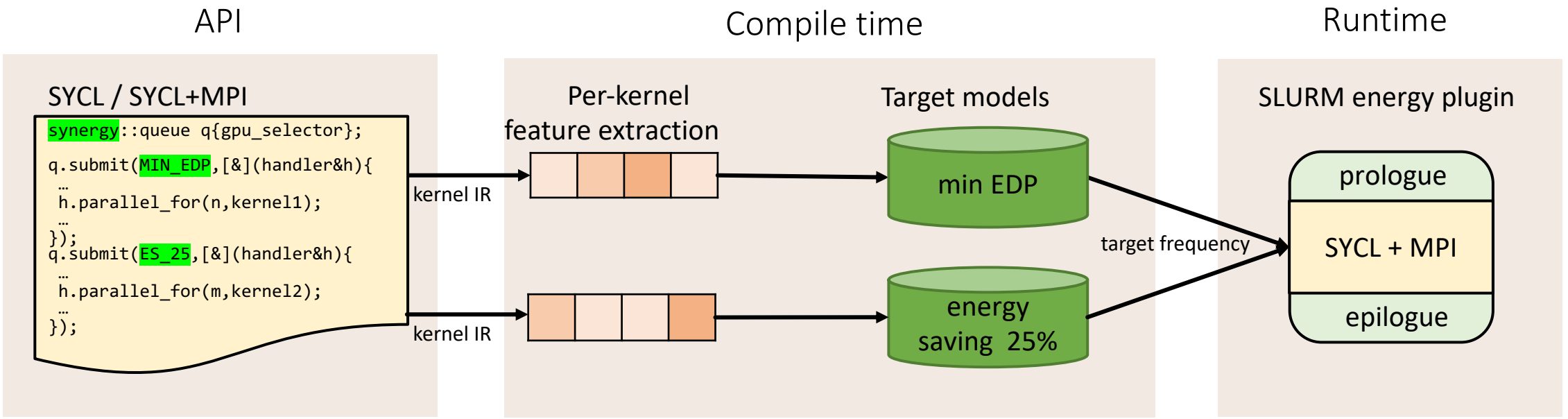



Performance-loss targets

New energy targets

- ES_x: the frequency configuration that delivers the x% relative energy savings
- PL_x: the frequency configuration that has x% relative performance loss
- Relative to the range [default, MIN_ENERGY]

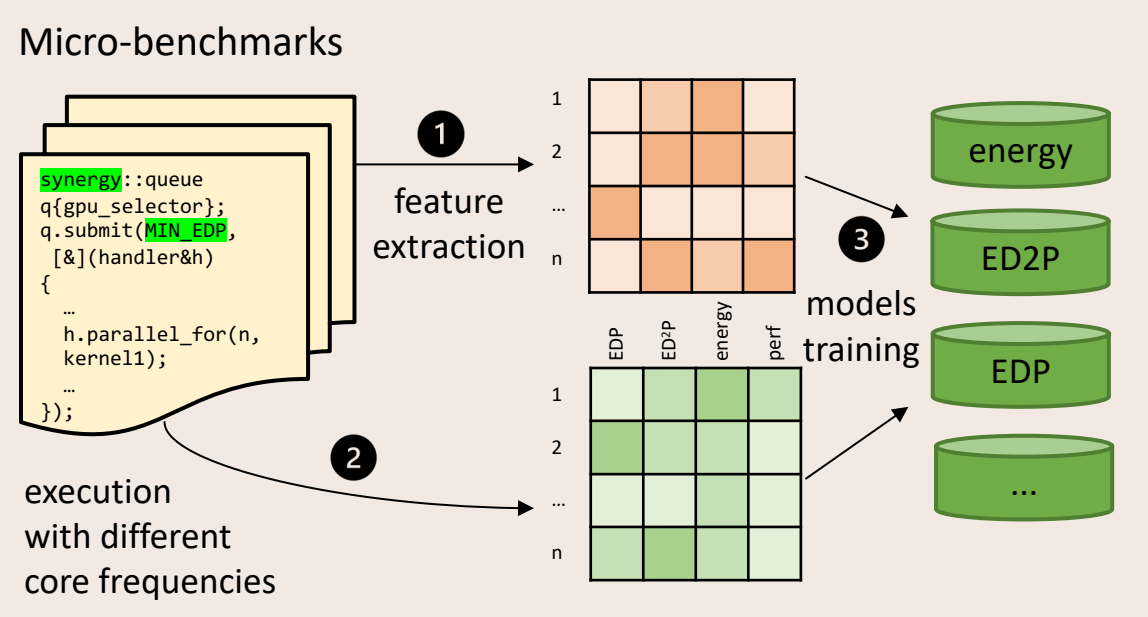
# Synergy Compilation



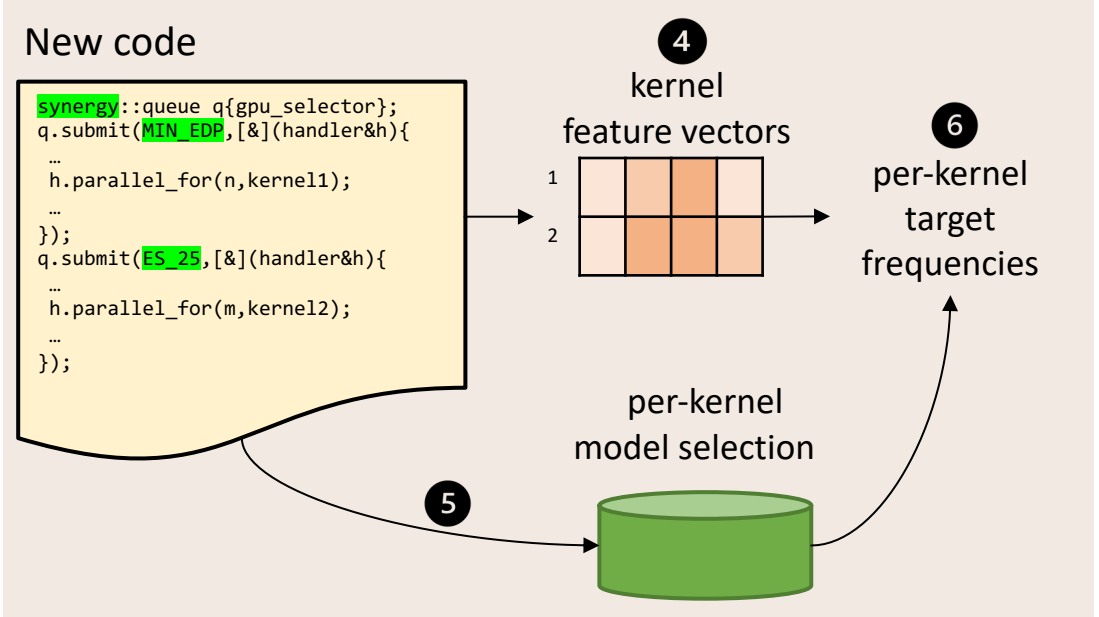

- Infer the optimal frequency configuration for user-defined energy target

# SYnergy Compilation: Energy Target Models

## Training



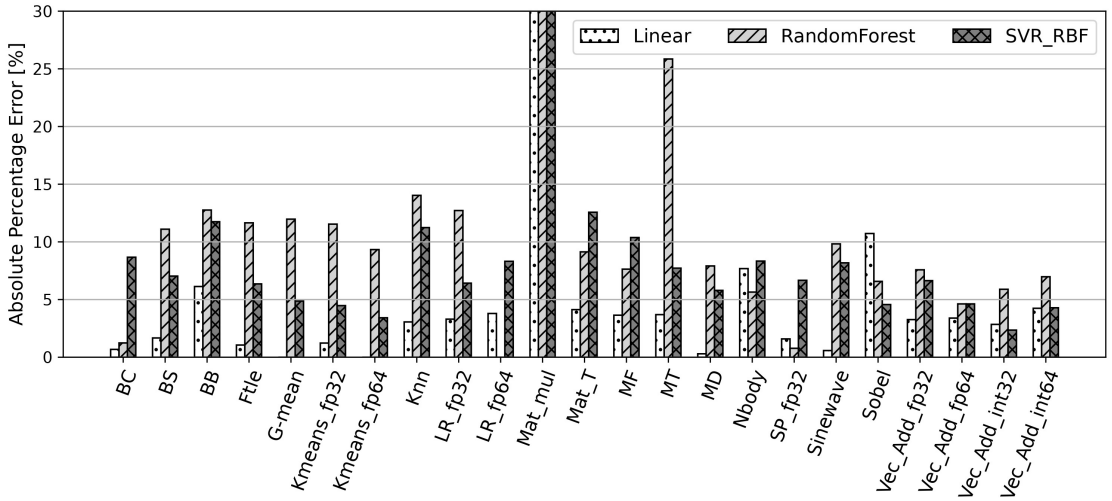

## Inference



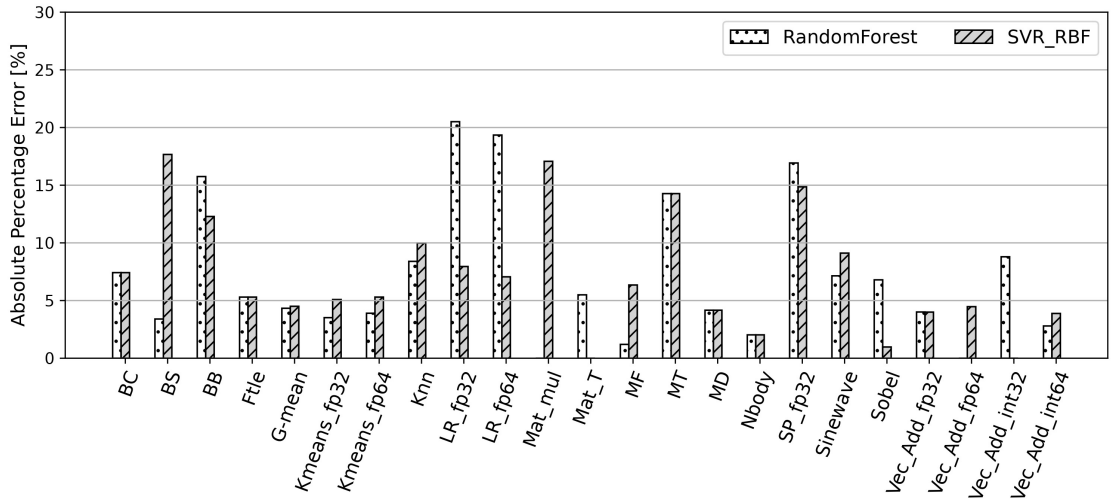

- Energy characterization depends on the code & target hardware
  - kernel is characterized by static code features extracted by a LLVM pass
- Energy model based on machine learning
  - Training on microbenchmarks, evaluation on 23 benchmarks (SYCL-Bench), leave-one-out cross validation, each target use different models

# Result: Prediction Error Analysis on Single Node

MIN_ED2P: frequency prediction error

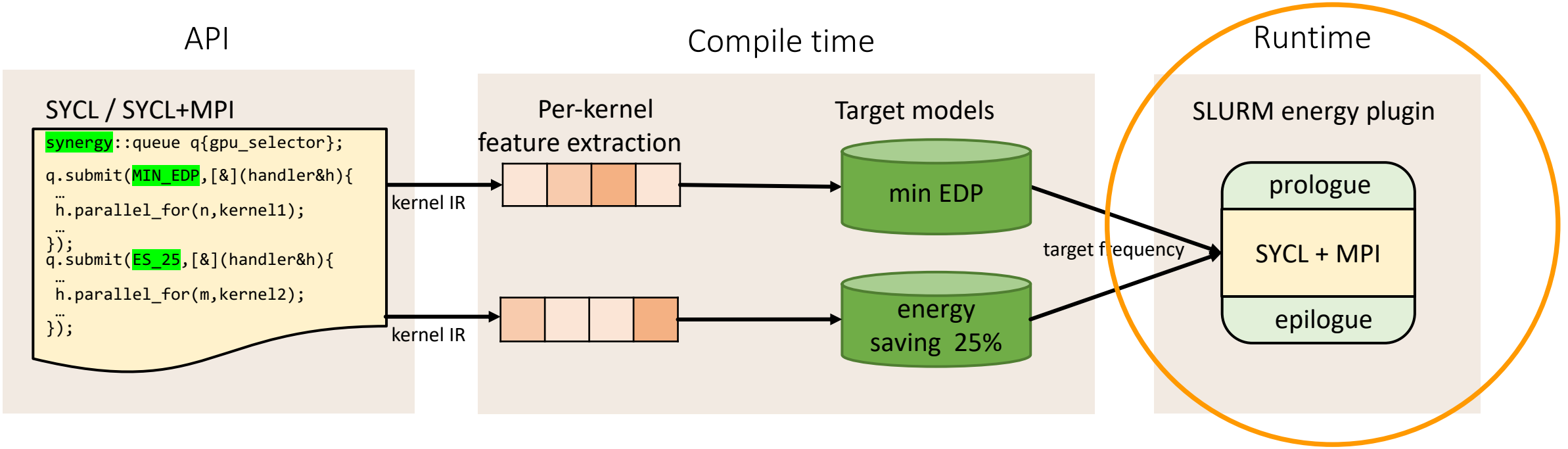



MIN_ENERGY: frequency prediction error




- Best for performance-related targets: Linear Regression
- Best for energy-related targets: Random Forest Regression

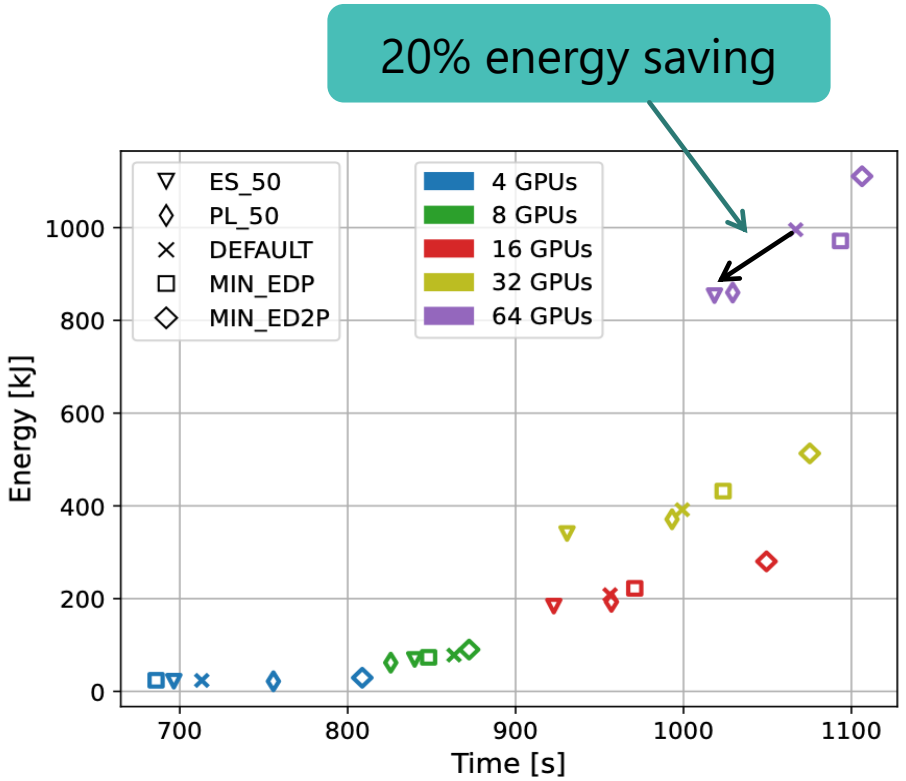
# SYnergy on Multi-nodes Systems



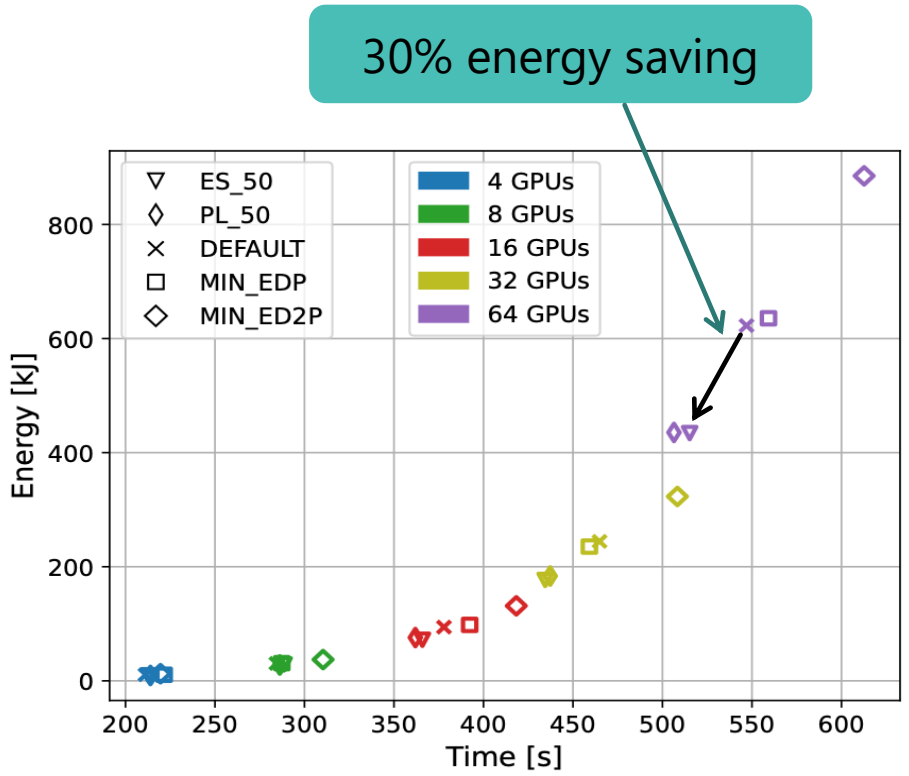

- SLURM plugin enables energy features on large clusters
  - SLURM job scheduler plugin extends the job execution policy with a specific prologue and epilogue, enabling energy-efficient computing on all devices in the job
  - evaluation on Marconi100 at CINECA, up to 64 GPU

# Strong Scaling Evaluation on Marconi100 / CINECA

- Applications: CloverLeaf and MiniWeather



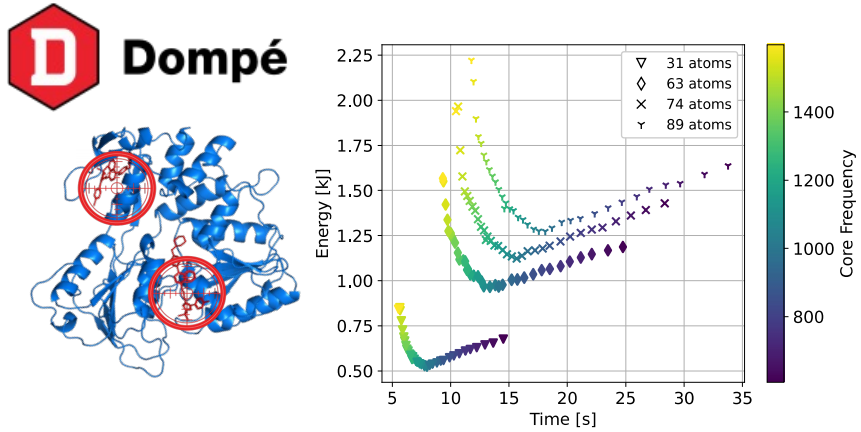

(a) CloverLeaf

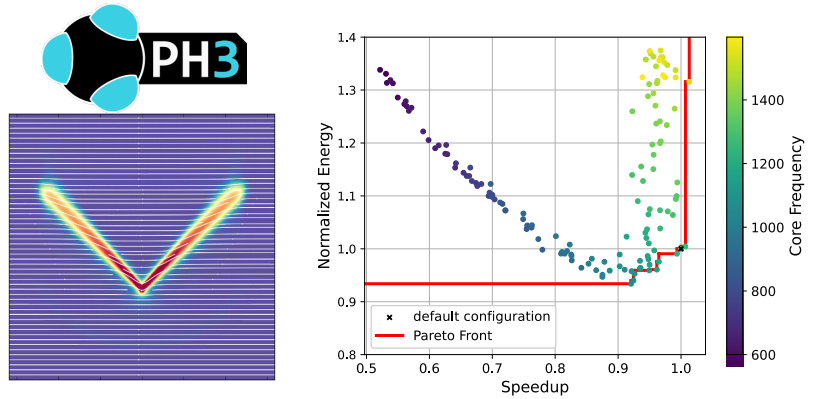



(b) MiniWeather

# SYnergy on Industrial Applications

- Two industry applications within the LIGATE EuroHPC project
- LiGen (Ligand Generator) by Dompé SpA
  - in silico simulator that runs on HPC systems
  - binding of 16.5 billion molecules can be virtually tested on a target protein in just 1h
- Cronos by PH3 Gmbh
  - a magnetohydrodynamics code developed for the solution of plasma-dynamical problems in astrophysics and space science
  - optimized for general hyperbolic equations, compressible hydrodynamics, special relativistic hydrodynamics



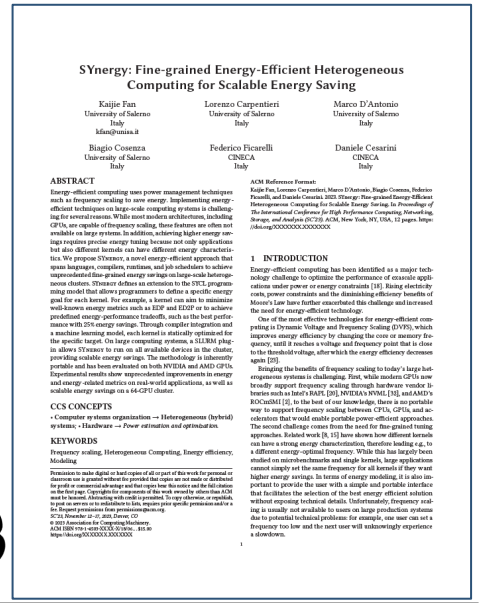




Lorenzo Carpentieri, Marco D'Antonio, Kaijie Fan, Luigi Crisci, Biagio Cosenza, Federico Ficarelli, Daniele Cesarini, Gianmarco Accordi, Davide Gadioli, Gianluca Palermo, Peter Thoman, Philip Salzmann, Philipp Gschwandtner, Markus Wippler, Filippo Marchetti, Daniele Gregori, Andrea Rosario Beccari: Domain-Specific Energy Modeling for Drug Discovery and Magnetohydrodynamics Applications. SC Workshops 2023: 1789-1800

# Summary & Conclusions

https://github.com/unisa-hpc/SYnergy

- SYnergy
  - SYCL interface for energy profiling and frequency scaling
  - Energy target and machine learning models
  - Energy-aware SLURM plug-in for energy scalability on multiple GPUs
- Next steps
  - Concurrent kernel executions for out-of-order queue
  - Target embedded systems



SYnergy: Fine-grained Energy-Efficient Heterogeneous Computing for Scalable Energy Saving

Kaijie Fan, University of Salerno, Italy, kfan@unisa.it
Lorenzo Carpentieri, University of Salerno, Italy
Marco D'Antonio, University of Salerno, Italy
Biagio Cosenza, University of Salerno, Italy
Federico Ficarelli, CINECA, Italy
Daniele Cesarini, CINECA, Italy

ABSTRACT

Energy-efficient computing uses power management techniques such as frequency scaling to save energy. Implementing energy-efficient techniques on large-scale computing systems is challenging for several reasons. While most modern architectures, including GPUs, are capable of frequency scaling, these features are often not available on large systems. In addition, achieving higher energy savings requires precise energy tuning because not only applications but also different kernels can have different energy characteristics. We propose SYnergy, a novel energy-efficient approach that spans languages, compilers, runtimes, and job schedulers to achieve unprecedented fine-grained energy savings on large-scale heterogeneous clusters. SYnergy defines an extension to the SYCL programming model that allows programmers to define a specific energy goal for each kernel. For example, a kernel can aim to minimize well-known energy metrics such as EDP and ED2P or to achieve predefined energy-performance tradeoffs, such as the best performance with 25% energy savings. Through compiler integration and a machine learning model, each kernel is statically optimized for the specific target. On large computing systems, a SLURM plug-in allows SYnergy to run on all available devices in the cluster, providing scalable energy savings. The methodology is inherently portable and has been evaluated on both NVIDIA and AMD GPUs. Experimental results show unprecedented improvements in energy and energy-related metrics on real-world applications, as well as scalable energy savings on a 64-GPU cluster.

CCS CONCEPTS

• Computer systems organization → Heterogeneous (hybrid) systems; • Hardware → *Power estimation and optimization*.

KEYWORDS

Frequency scaling, Heterogeneous Computing, Energy efficiency, Modeling

ACM Reference Format:
Kaijie Fan, Lorenzo Carpentieri, Marco D'Antonio, Biagio Cosenza, Federico Ficarelli, and Daniele Cesarini. 2023. SYnergy: Fine-grained Energy-Efficient Heterogeneous Computing for Scalable Energy Saving. In *The International Conference for High Performance Computing, Networking, Storage, and Analysis (SC'23), November 12–17, 2023, Denver, CO.* ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Energy-efficient computing has been identified as a major technology challenge to optimize the performance of exascale applications under power or energy constraints [18]. Rising electricity costs, power constraints and the diminishing efficiency benefits of Moore's Law have further exacerbated this challenge and increased the need for energy-efficient technology.

One of the most effective technologies for energy-efficient computing is Dynamic Voltage and Frequency Scaling (DVFS), which improves energy efficiency by changing the core or memory frequency, until it reaches a voltage and frequency point that is close to the threshold voltage, after which the energy efficiency decreases again [20].

Bringing the benefits of frequency scaling to today's large heterogeneous systems is challenging. First, while modern GPUs now broadly support frequency scaling through hardware vendor libraries such as Intel's RAPL [20], NVIDIA's NVML [31], and AMD's ROCm SMI [2], to the best of our knowledge, there is no portable way to support frequency scaling between CPUs, GPUs, and accelerators that would enable portable power-efficient approaches. The second challenge comes from the need for fine-grained tuning approaches. Related work [8, 15] have shown how different kernels can have a strong energy characterization, therefore leading e.g., to a different energy-optimal frequency. While this has largely been studied on microbenchmarks and single kernels, large applications cannot simply set the same frequency for all kernels if they want higher energy savings. In terms of energy modeling, it is also important to provide the user with a simple and portable interface that facilitates the selection of the best energy efficient solution without exposing technical details. Unfortunately, frequency scaling is usually not available to users on large production systems due to potential technical problems: for example, one user can set a frequency too low and the next user will unknowingly experience a slowdown.

SC23 DENVER NOV 12–17

Fan, Carpentieri, D'Antonio, Cosenza, Ficarelli, Cesarini: SYnergy: Fine-grained Energy-Efficient Heterogeneous Computing for Scalable Energy Saving. SC 2023

This project has received funding from the European High-Performance Computing Joint Undertaking Joint Undertaking (JU) under grant agreement No 956137. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Italy, Sweden, Austria, Czech Republic, Switzerland.

UNIVERSITÀ DEGLI STUDI DI SALERNO