.AMPERE

A Model-driven development framework for highly Parallel and EneRgy-Efficient computation supporting multi-criteria optimisation

STATIC ANALYSIS OF NON-FUNCTIONAL REQUIREMENTS
Energy efficiency

AMPERE Final Event Webinar

Sergio Mazzola — ETH Zürich
27 June 2023

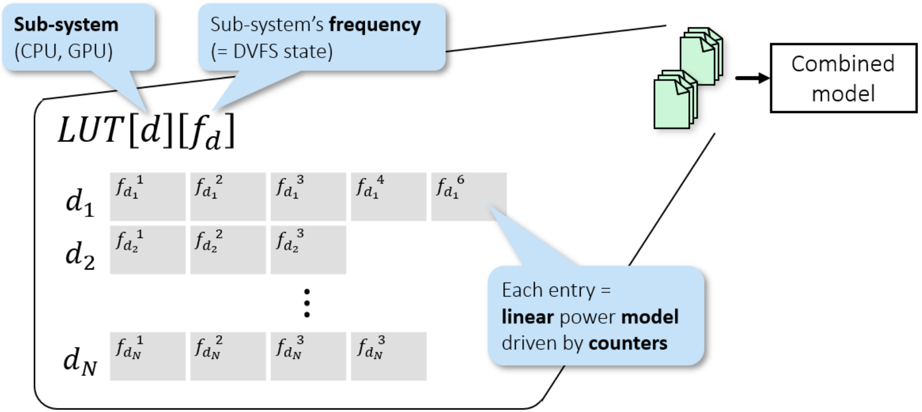
The AMPERE project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871669

Performance-counter-based power models – Why?

- **AMPERE target platforms: modern high-performance systems**
  - Heterogeneity, parallelism
  - Dynamic voltage & frequency scaling (DVFS)
- **Analog power meters: slow, no introspection**
  - **Perf. counters**: close to the digital hw domain, fast, reliable
  - No integration required, cheap-to-use
- **Architecture-agnostic, data-driven parameters selection**
  - Makes it flexible and platform-independent
- **Support for DVFS and arbitrary granularity**
  - With high accuracy and low overhead

How does all this come together?

- **System-level power model = Look-up Table**
  - One entry for each **sub-system**, at each **frequency**
  - Each entry = linear power model, driven by counters
- **Low overhead – still supporting heterogeneity and DVFS**
- **Power estimates used to compute energy consumption**



Sub-system (CPU, GPU)

Sub-system's **frequency** (= DVFS state)

$LUT[d][f_d]$

d_1	$f_{d_1}{}^1$	$f_{d_1}{}^2$	$f_{d_1}{}^3$	$f_{d_1}{}^4$	$f_{d_1}{}^6$
d_2	$f_{d_2}{}^1$	$f_{d_2}{}^2$	$f_{d_2}{}^3$		
			$\vdots$		
d_N	$f_{d_N}{}^1$	$f_{d_N}{}^2$	$f_{d_N}{}^3$	$f_{d_N}{}^3$	

Each entry = **linear** power **model** driven by **counters**

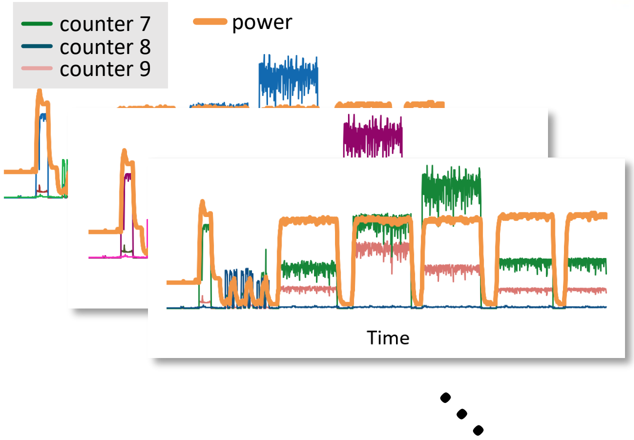
Combined model

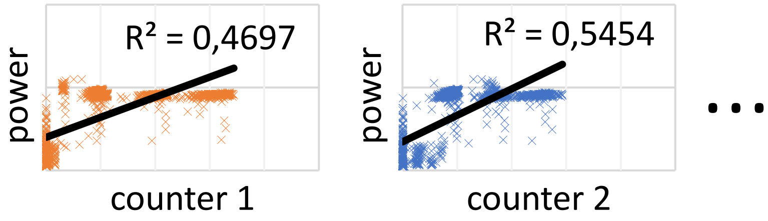
Our holistic model building methodology

1. **Workloads selection**
   - Coverage of all sub-systems
   - Broad coverage of each sub-system’s behaviors
2. **One-time platform characterization**
   - Autonomous, statistical selection of the **best hardware counters** to use as **model parameters**, for a **given platform**
3. **Training and building of the LUT**
   - Linear power models
     - Per sub-system (CPU, GPU, …)
     - Per sub-system’s frequency

Platform characterization

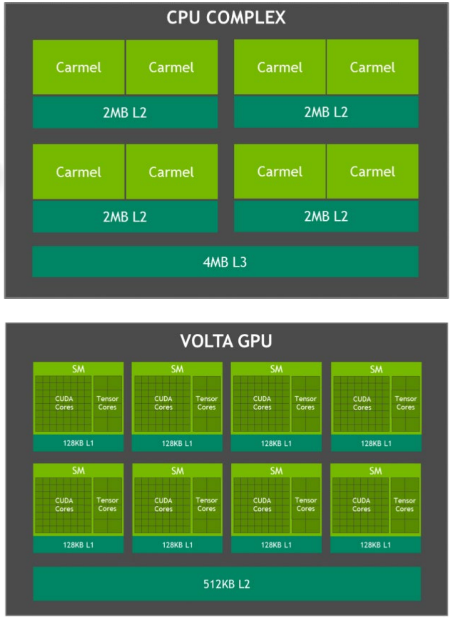
- **Statistical data-driven approach**
  - One time per platform
  - No architectural knowledge required, flexible
- **Procedure**
  1. Profile (all) counters & power
  2. Compute PCC
     Person Correlation Coefficient
  3. Select best counters
     Fine-tune trade-off between number of counters (i.e., **model overhead**) and **accuracy**





Experimental setup – NVIDIA Jetson AGX Xavier

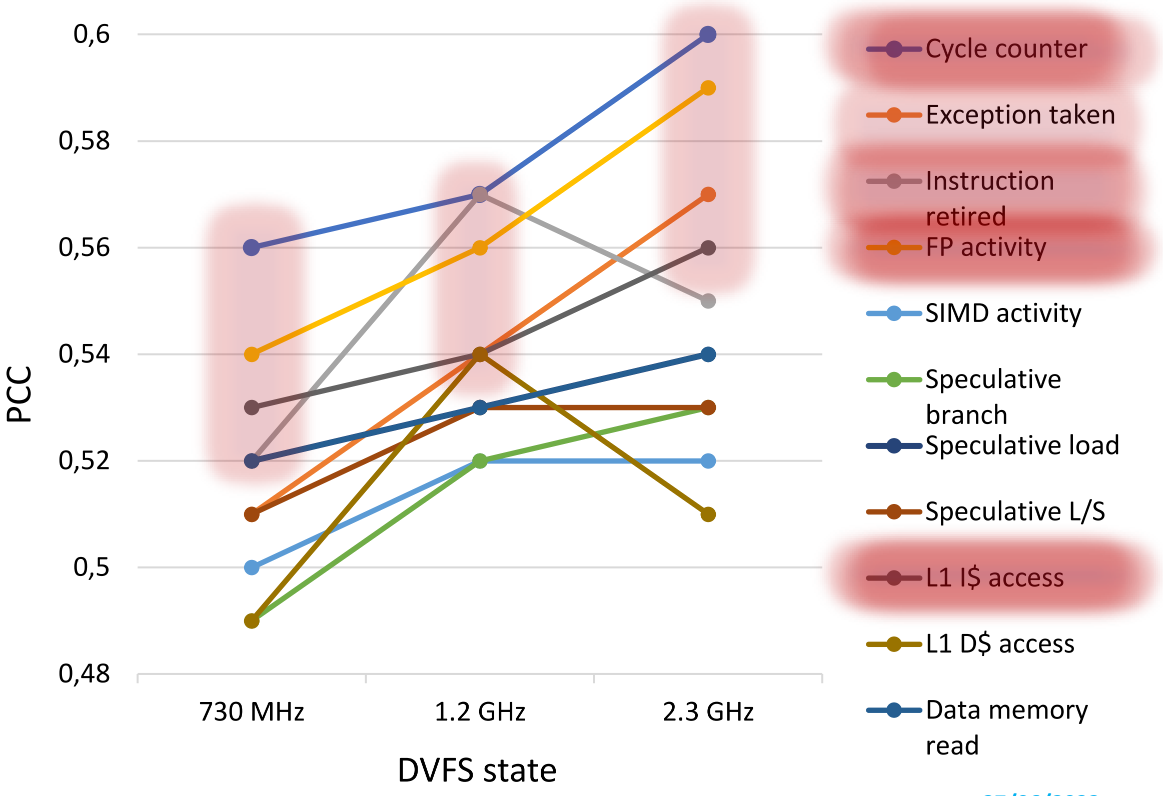
- **8-core 64-bit ARM SoC**
  - Per-cluster DVFS
  - 29 nominal frequencies
    - 115 MHz – 2.3 GHz
- **512-core NVIDIA Volta GPU**
  - 14 nominal frequencies
    - 115 MHz – 1.4 GHz
- **2 on-board power monitors (INA3221)**
  - Analog current sensors
  - Useful to build better models



Source: https://developer.nvidia.com/blog

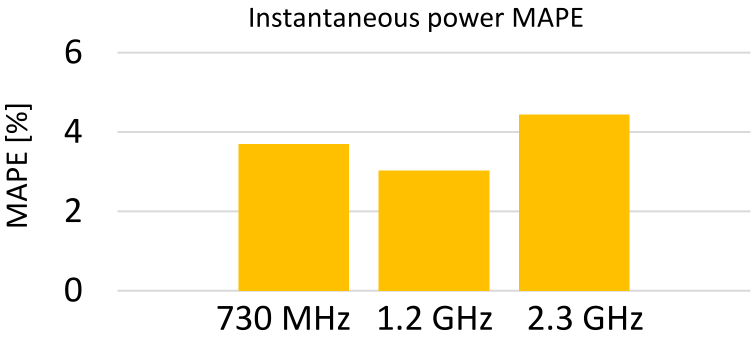
Example case study: CPU platform characterization

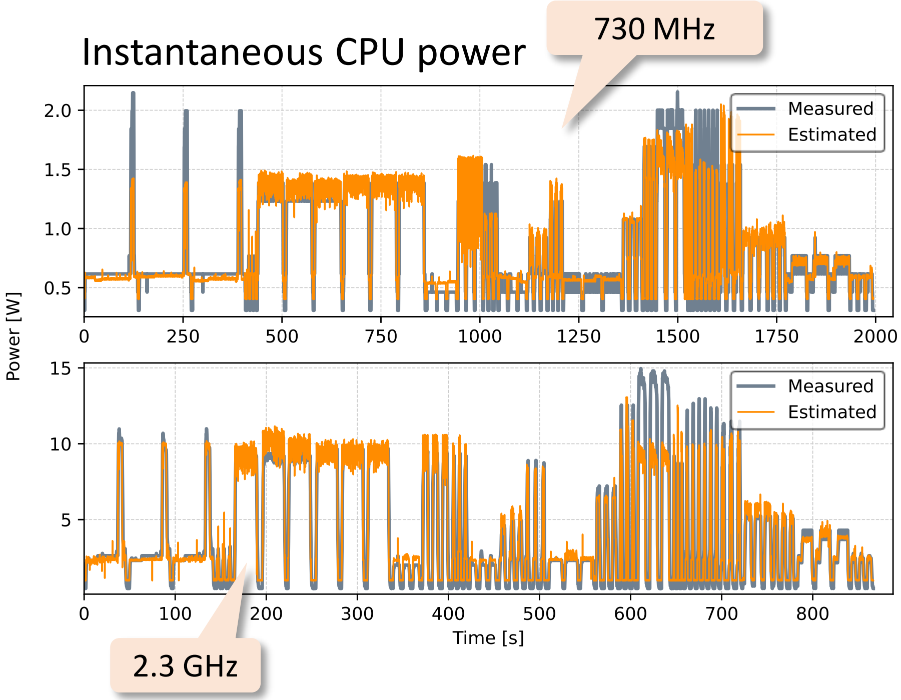
- **Fixed clock cycle counter**
  Always available by default
- **Best 3 PMCs @ each frequency**
- **Per-cluster DVFS: same counter for each core**



Example case study: CPU model validation

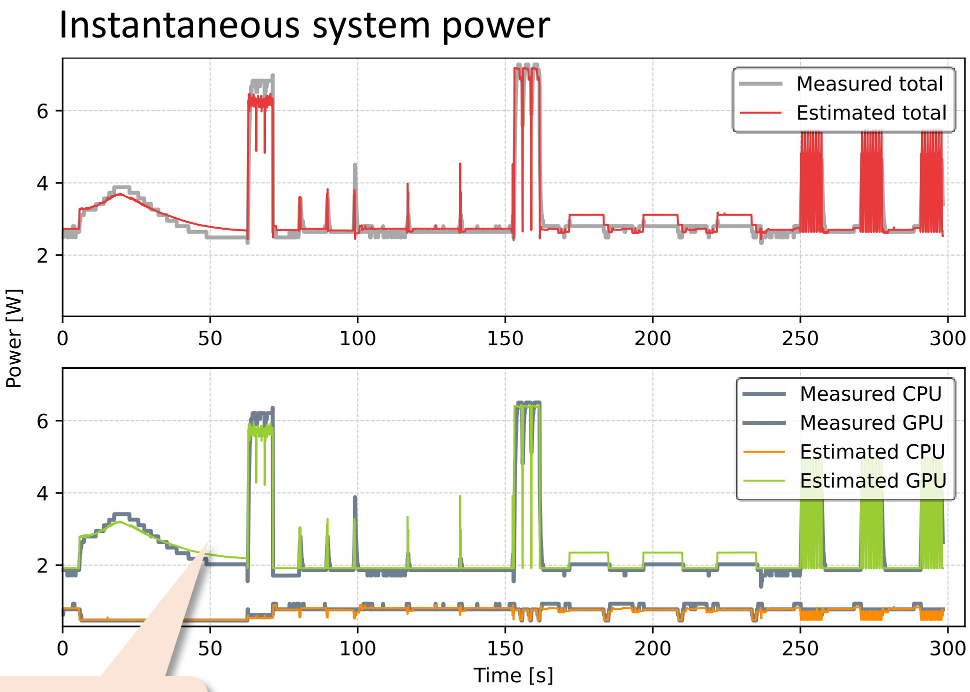
- **Power tracked overtime**
- **Instantaneous avg power error**
  ~4% over all frequencies
- **Total energy estimation error**
  ~4% over all frequencies





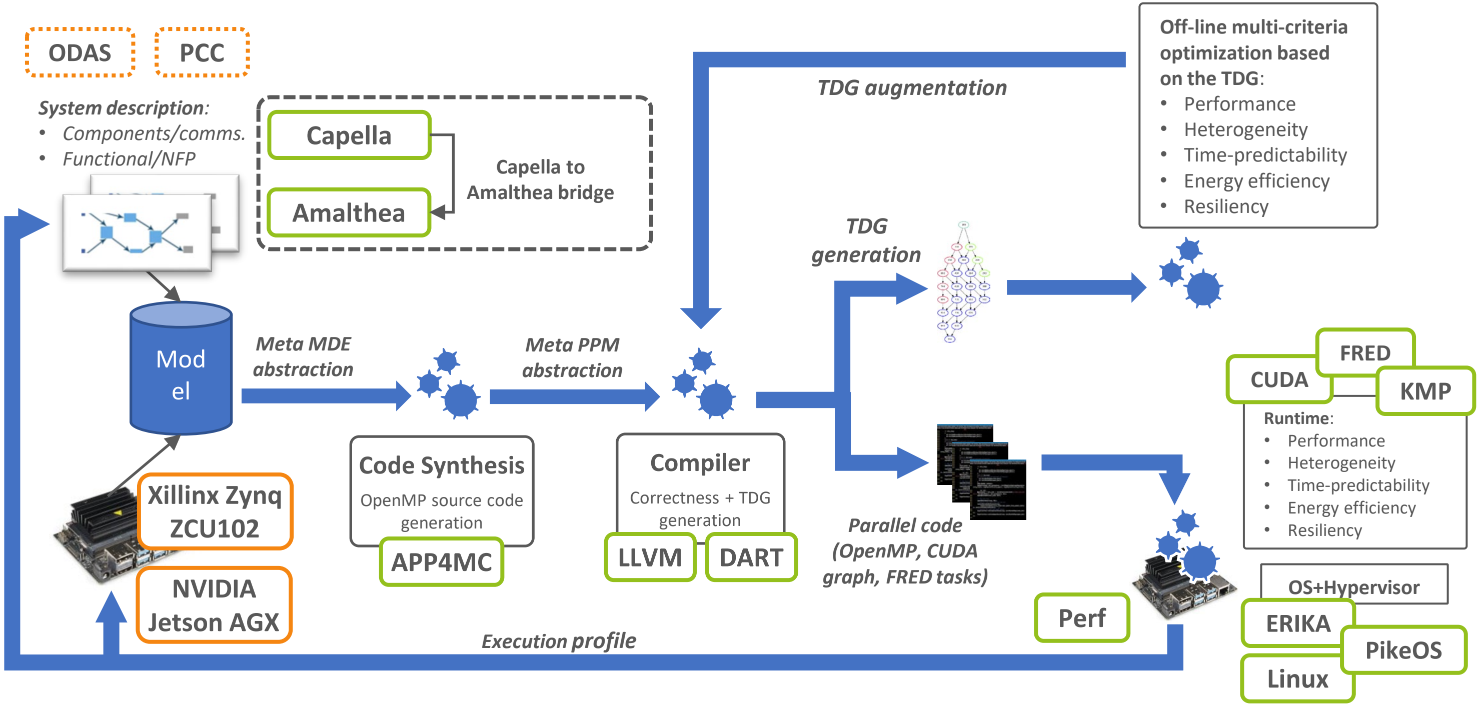
Combined system-level model validation

- Instantaneous power avg error = ~7.5%
- Total energy estimation error = ~1.3%

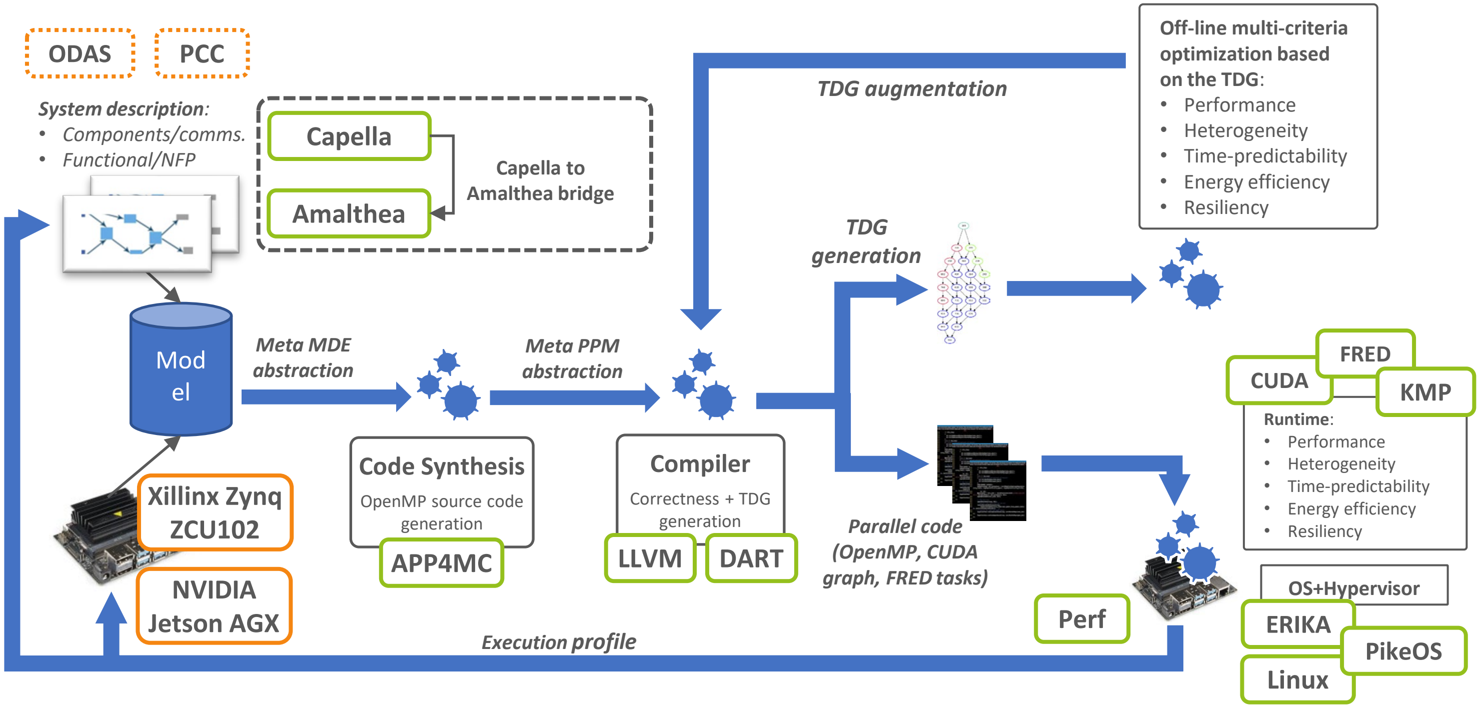


CPU: 1.2 GHz
GPU: 829 MHz

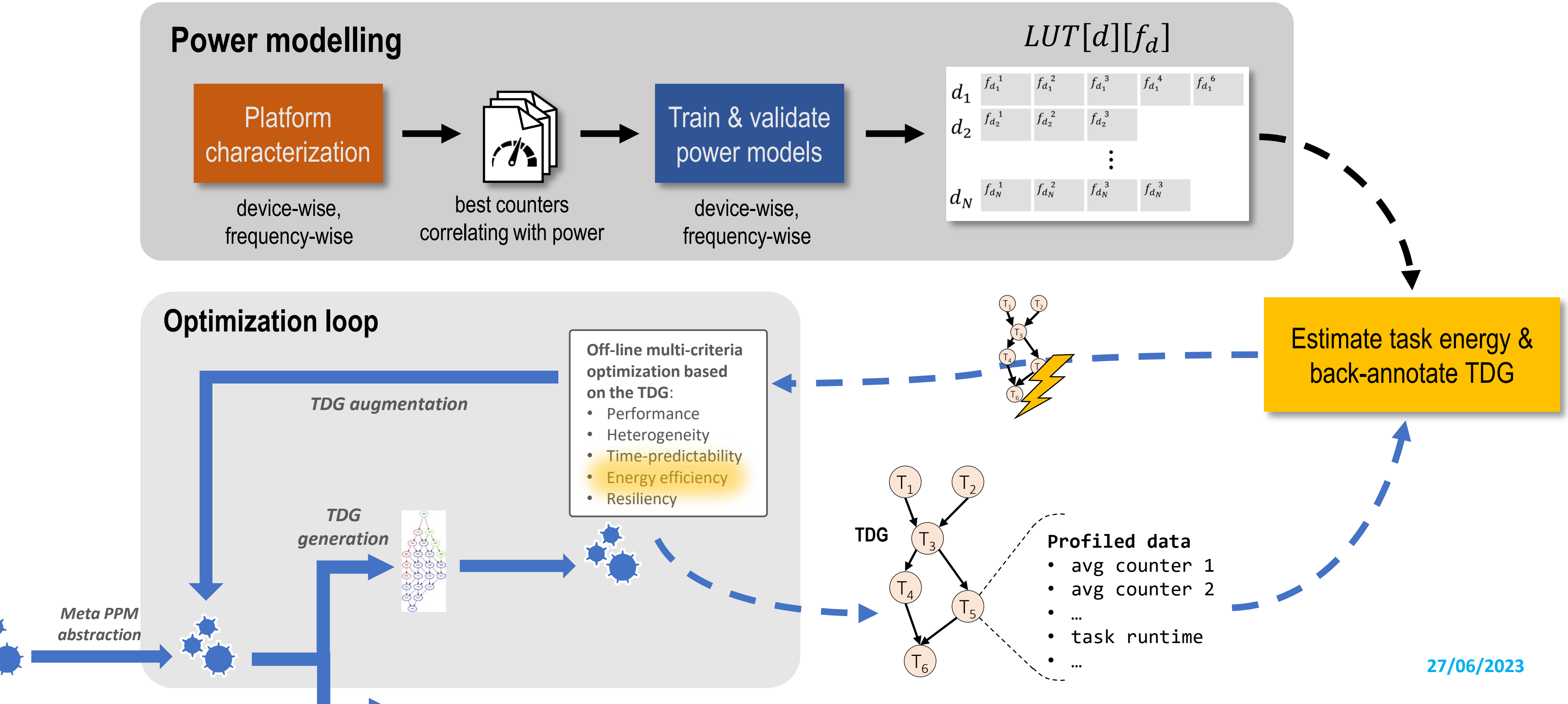
Energy modelling in the multi-criteria optimization



Energy modelling in the multi-criteria optimization



Energy modelling in the multi-criteria optimization



Power modelling

Platform characterization

device-wise, frequency-wise

best counters correlating with power

Train & validate power models

device-wise, frequency-wise

$LUT[d][f_d]$

d_1	$f_{d_1}{}^1$	$f_{d_1}{}^2$	$f_{d_1}{}^3$	$f_{d_1}{}^4$	$f_{d_1}{}^6$
d_2	$f_{d_2}{}^1$	$f_{d_2}{}^2$	$f_{d_2}{}^3$		
			$\vdots$		
d_N	$f_{d_N}{}^1$	$f_{d_N}{}^2$	$f_{d_N}{}^3$	$f_{d_N}{}^3$	

Optimization loop

Off-line multi-criteria optimization based on the TDG:

- Performance
- Heterogeneity
- Time-predictability
- Energy efficiency
- Resiliency

TDG augmentation

TDG generation

Meta PPM abstraction

Estimate task energy & back-annotate TDG

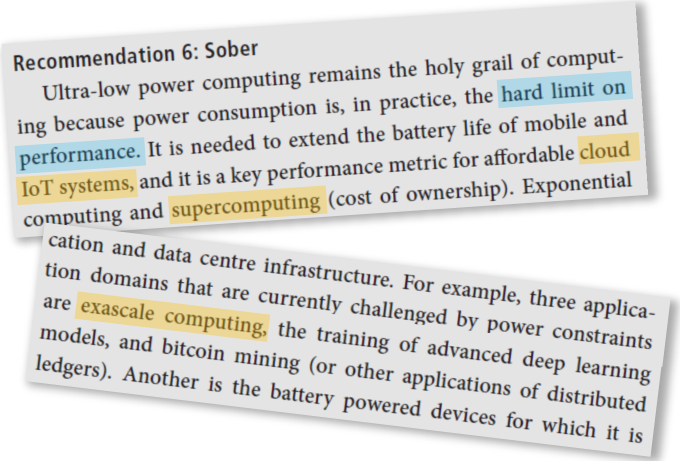
Profiled data

- avg counter 1
- avg counter 2
- …
- task runtime
- …

Why energy?

- **Green, sustainable computing**
- **Limited energy budget**
  - Application-dependent
  - AMPERE use cases (automotive: battery-powered)
  - Optimize energy consumption and monitor it
- **We are in the post-Dennard-scaling era**
  - Energy efficiency is the cost-effective way to get higher performance

Duranton, Marc, et al. "**HiPEAC Vision 2021**: high performance embedded architecture and compilation." (2021).



Recommendation 6: Sober

Ultra-low power computing remains the holy grail of computing because power consumption is, in practice, the hard limit on performance. It is needed to extend the battery life of mobile and IoT systems, and it is a key performance metric for affordable cloud computing and supercomputing (cost of ownership). Exponential

cation and data centre infrastructure. For example, three application domains that are currently challenged by power constraints are exascale computing, the training of advanced deep learning models, and bitcoin mining (or other applications of distributed ledgers). Another is the battery powered devices for which it is