CONTROL PULP - INTRO

SCALABLE RISC-V POWER CONTROLLER PLATFORM FOR HPC PROCESSORS

Andrea Bartolini, Giovanni Bambini, University of Bologna

OUTLINE

- **Power Management in HPC**
- ControlPulp Project: an open-source hardware/software RISC-V controller
- ControlPULP HIL Co-Desing Framework
- Co-simulation Demo
- QnA

POWER MANAGEMENT IN HPC: INTRO

Recent Design Challenges:

- **End** of Dennard's scaling law (slow down in transistors scaling)
- **Power and Thermal** challenges of modern **Multi-core** and **Many-core** designs
- Improving **energy efficiency** of the whole system
- **Security** concerns

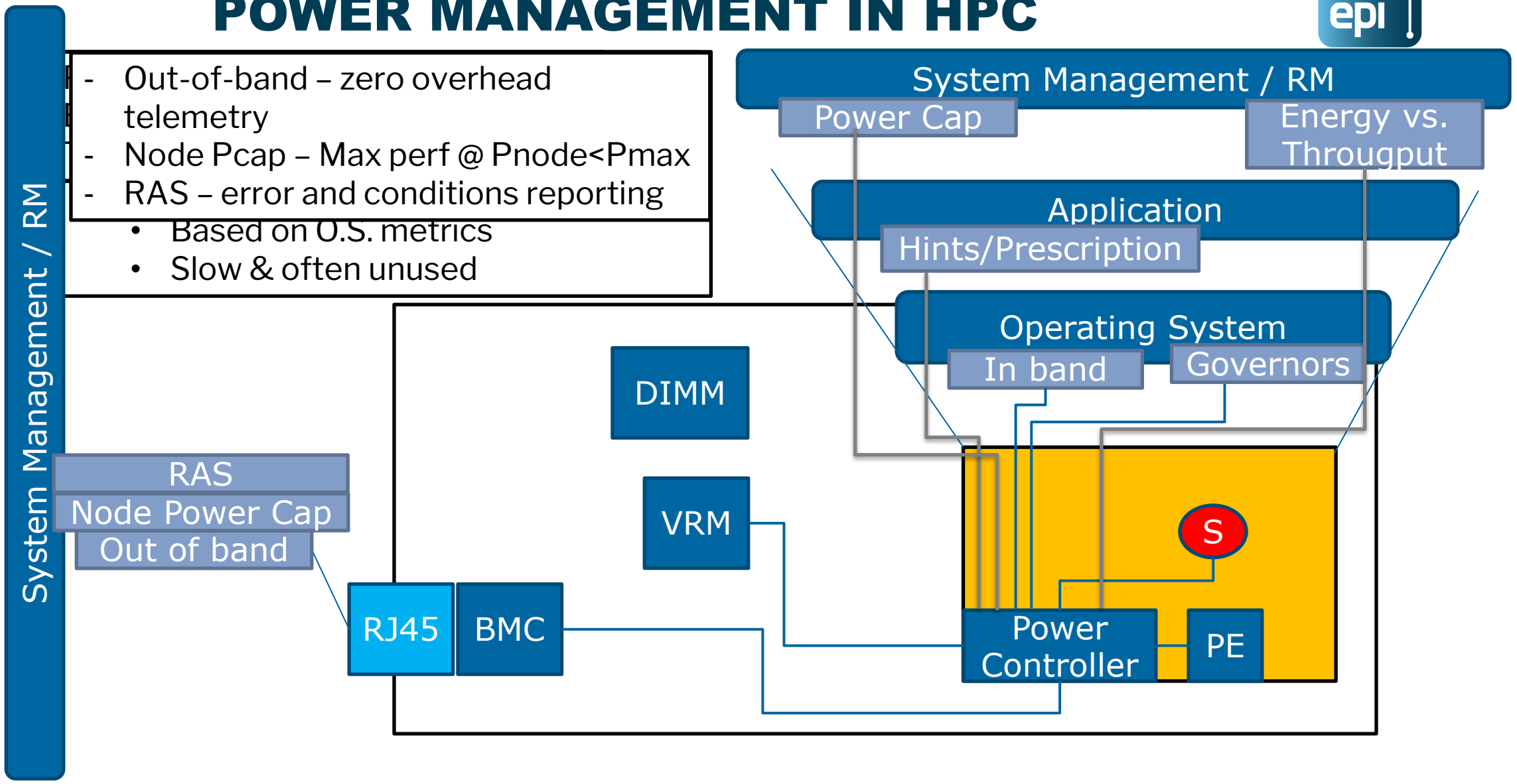
POWER MANAGEMENT IN HPC: INTRO

Design Choices:

- Specialization and **Custom chips** (Google, Apple, Amazon, …)
- **Heterogeneous** computing units on a single die
- **3D** chips
- Choice of an **optimal** operating point
- Advanced embedded **controller** (PMS)

POWER MANAGEMENT IN HPC

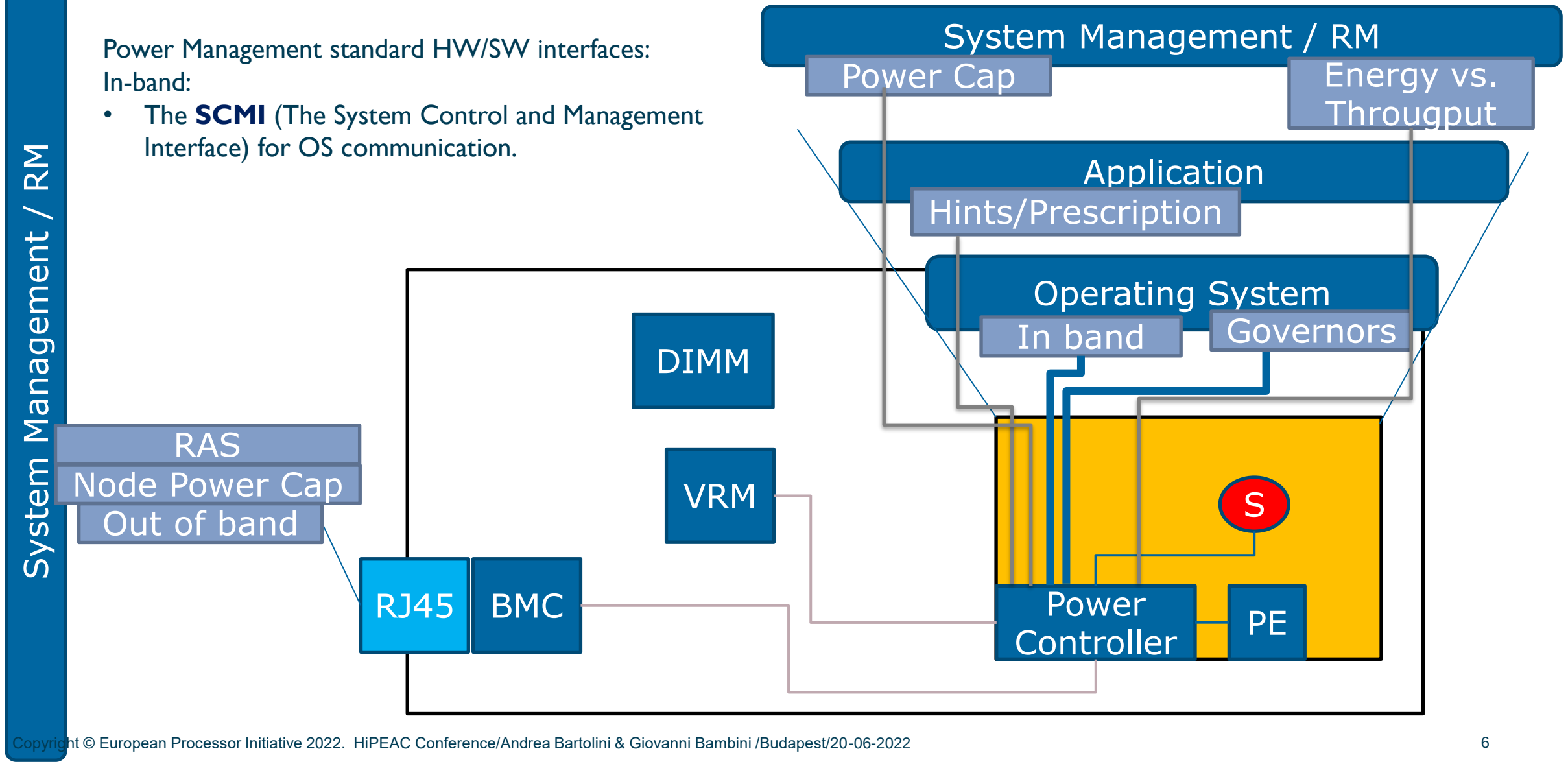
- Out-of-band – zero overhead telemetry
- Node Pcap – Max perf @ Pnode<Pmax
- RAS – error and conditions reporting
  - Based on O.S. metrics
  - Slow & often unused



POWER MANAGEMENT IN HPC

Power Management standard HW/SW interfaces:
In-band:

- The **SCMI** (The System Control and Management Interface) for OS communication.



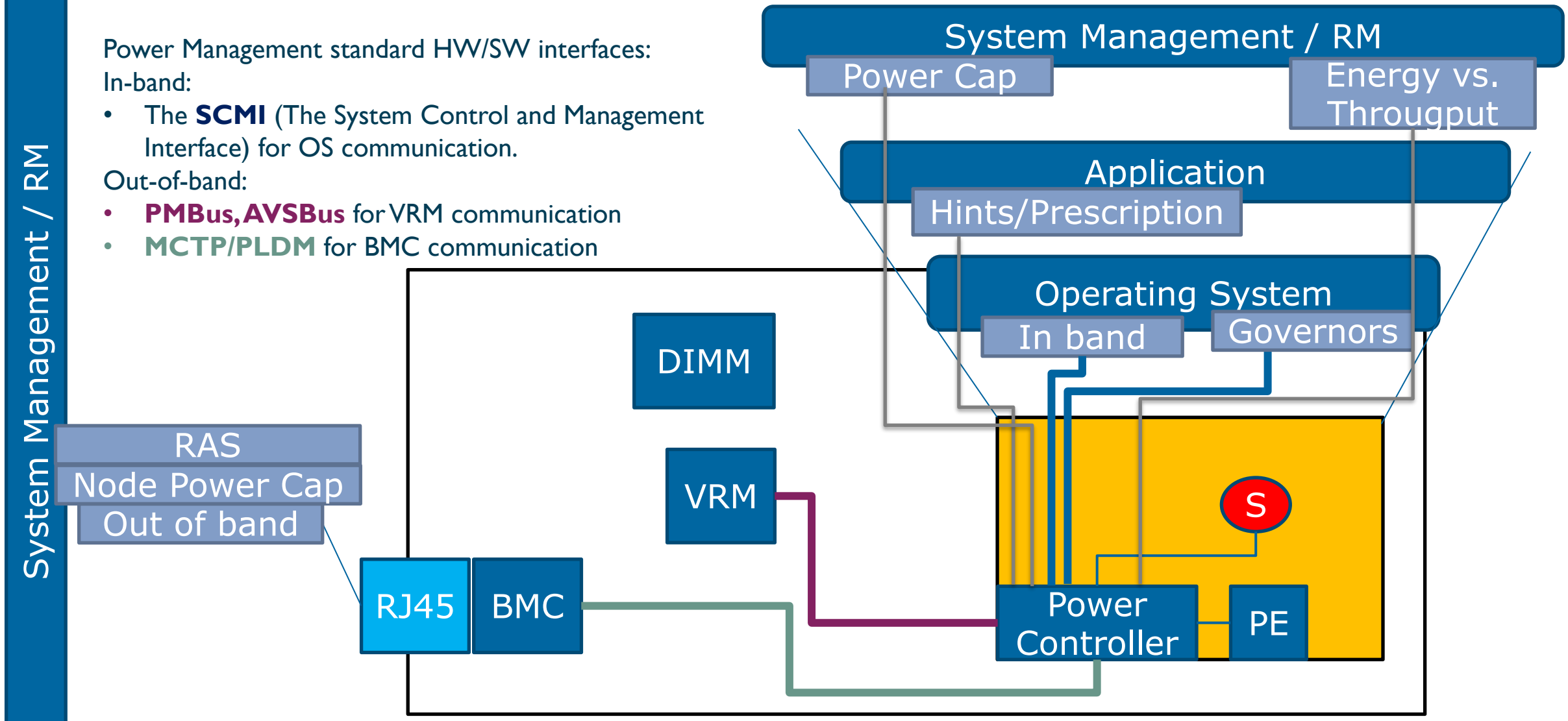
POWER MANAGEMENT IN HPC

Power Management standard HW/SW interfaces:
In-band:

- The **SCMI** (The System Control and Management Interface) for OS communication.

Out-of-band:

- **PMBus, AVSBus** for VRM communication
- **MCTP/PLDM** for BMC communication



OUTLINE

- Power Management in HPC
- **ControlPulp Project: an open-source hardware/software RISC-V controller**
- ControlPULP HIL Co-Desing Framework
- Co-simulation Demo
- QnA

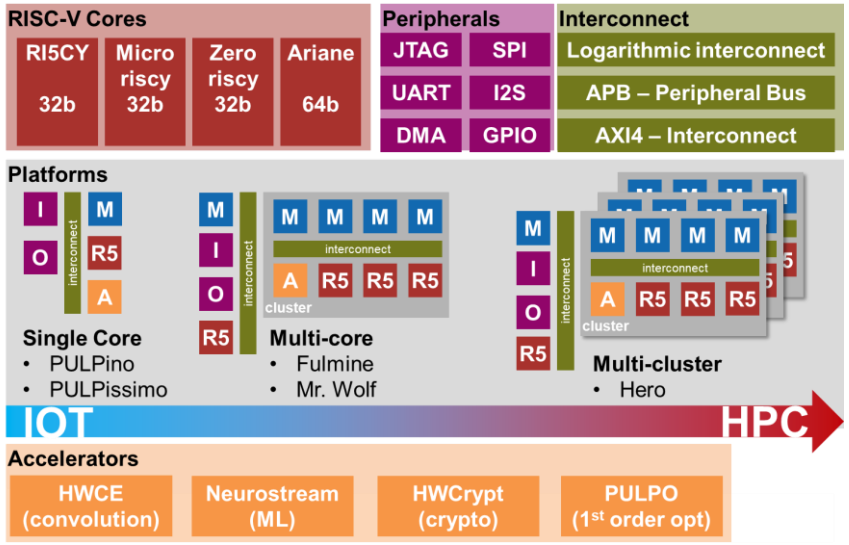
CONTROLPULP PROJECT: ARCHITECTURE

- PULP[1]-based design
- **Scalable architecture**:
  - **Multi-core** cluster with private FPU, up to float16 and bfloat precision
  - RISC-V **fast-interrupt controller: CLIC**
  - DMA for 2-D strided access from PVT sensor registers
- **Industry standard power management interfaces**:
  - **PMBUS**: Voltage Regulators control - slow/multi
  - **AVSBUS**: Voltage Regulators control - fast/p2p
  - **SPI**: Inter-socket communication (Multi ControlPULP)
  - **ACPI/MCTP**: Motherboard/BMC interface (OpenBMC)

Out-Of-Band

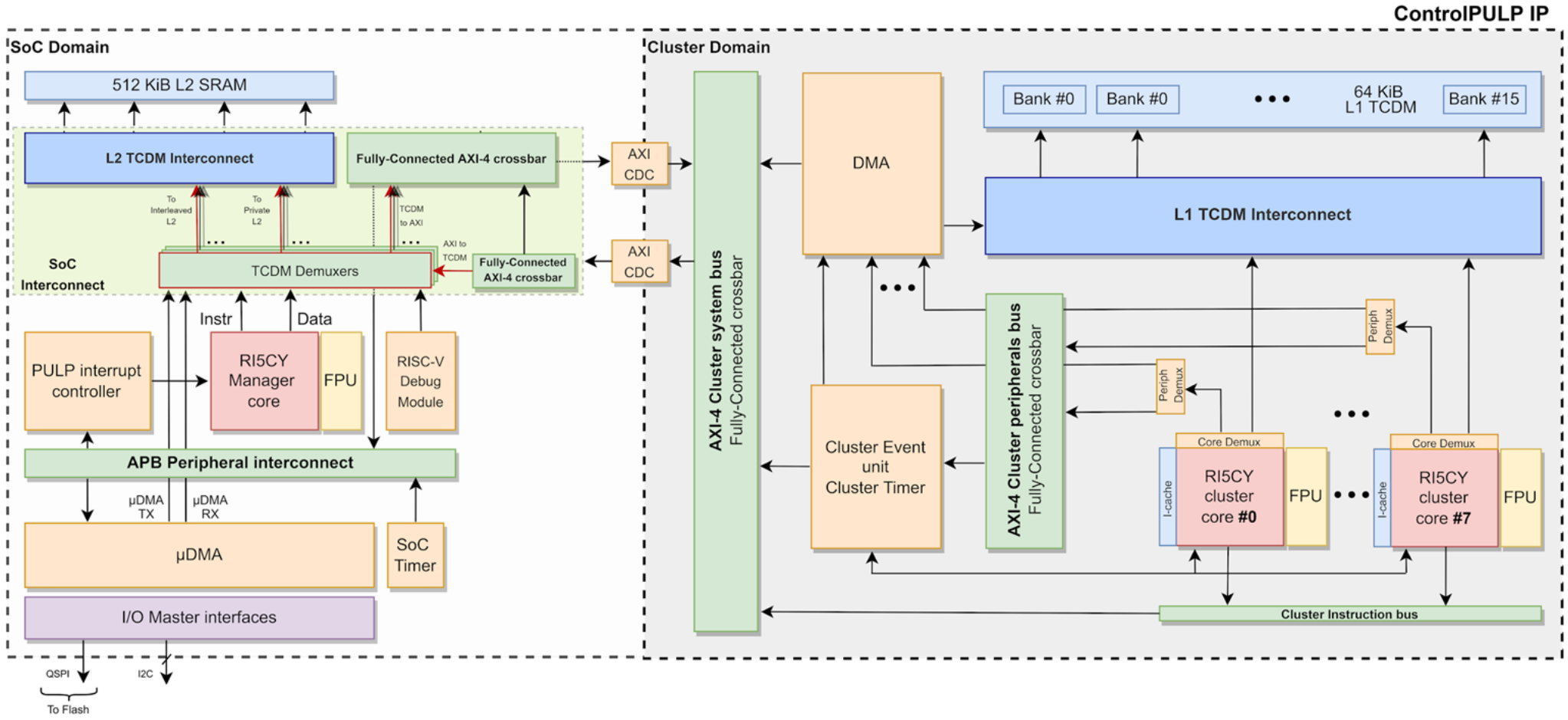
  - **SCMI**: OS PM governors and telemetry

In-Band

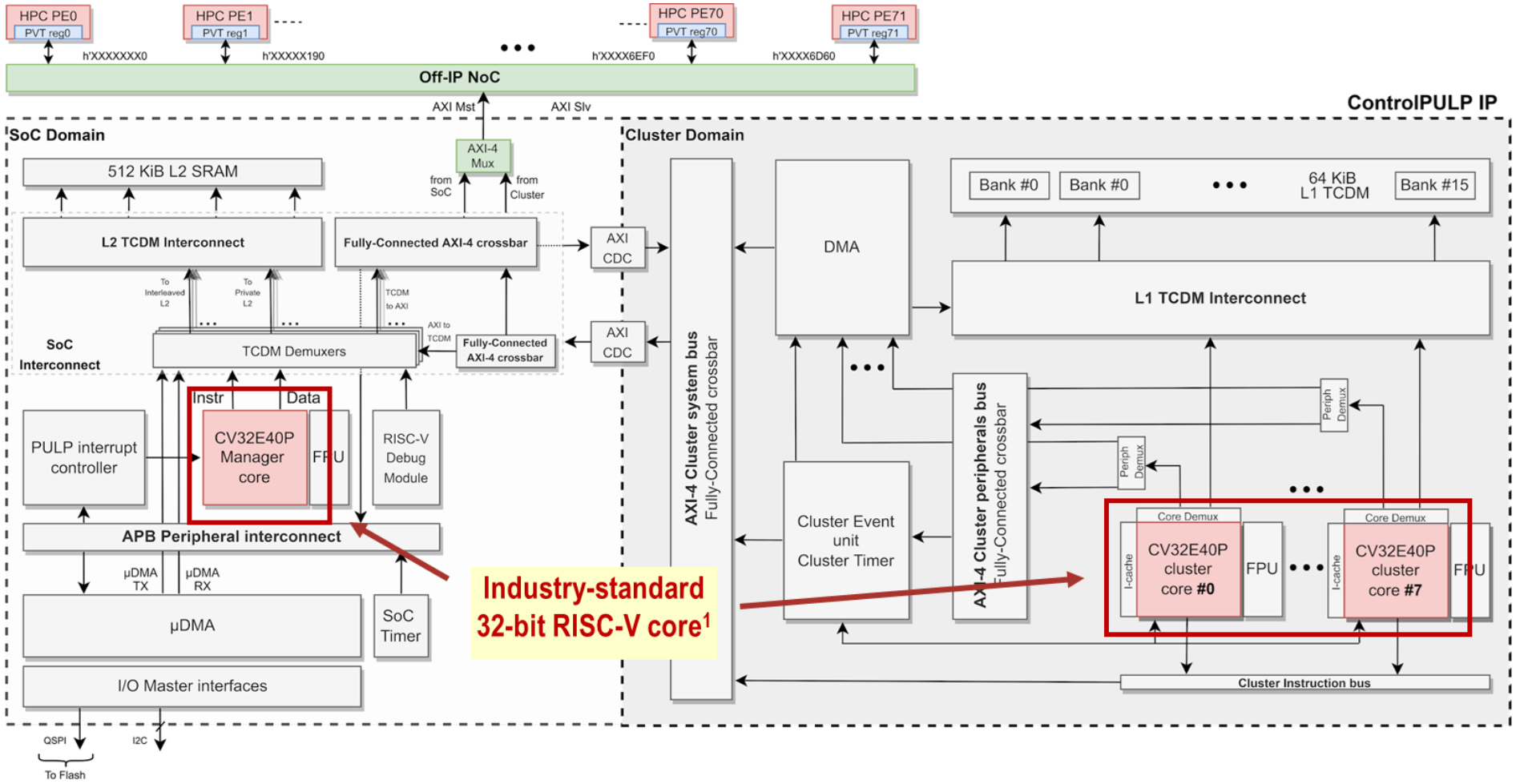


[1] https://github.com/pulp-platform/pulp

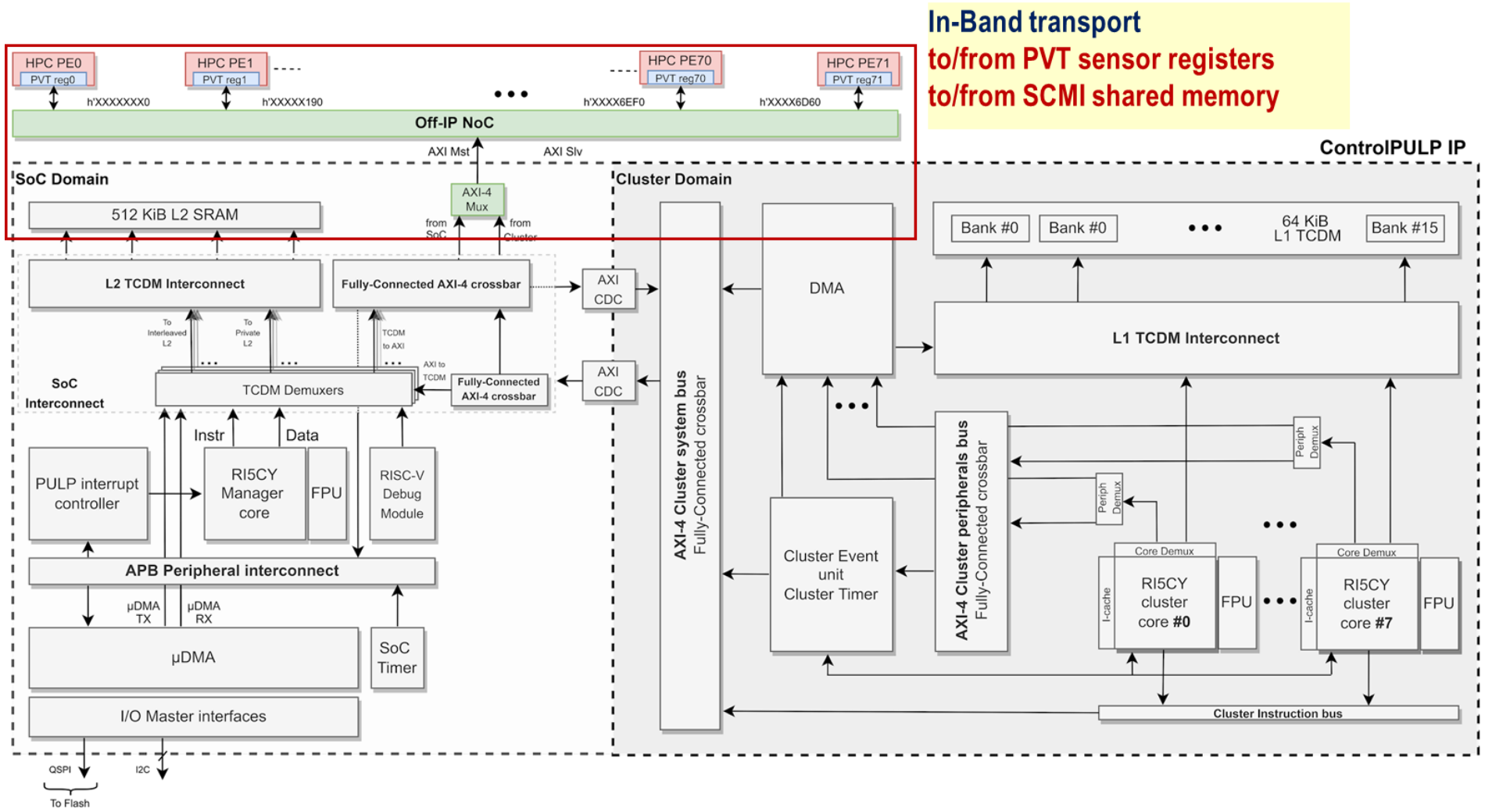
CONTROLPULP PROJECT: ARCHITECTURE



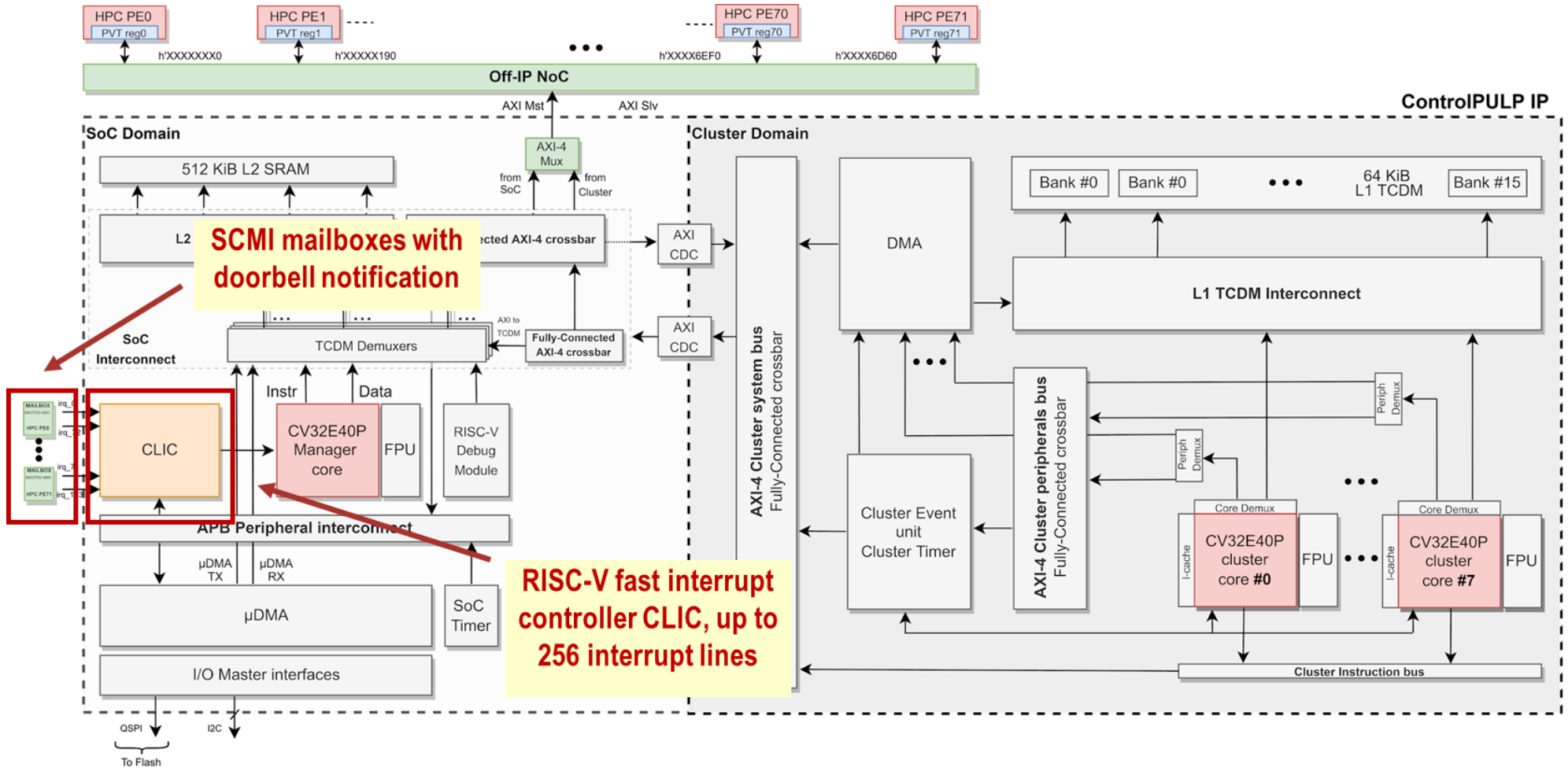
CONTROLPULP PROJECT: ARCHITECTURE



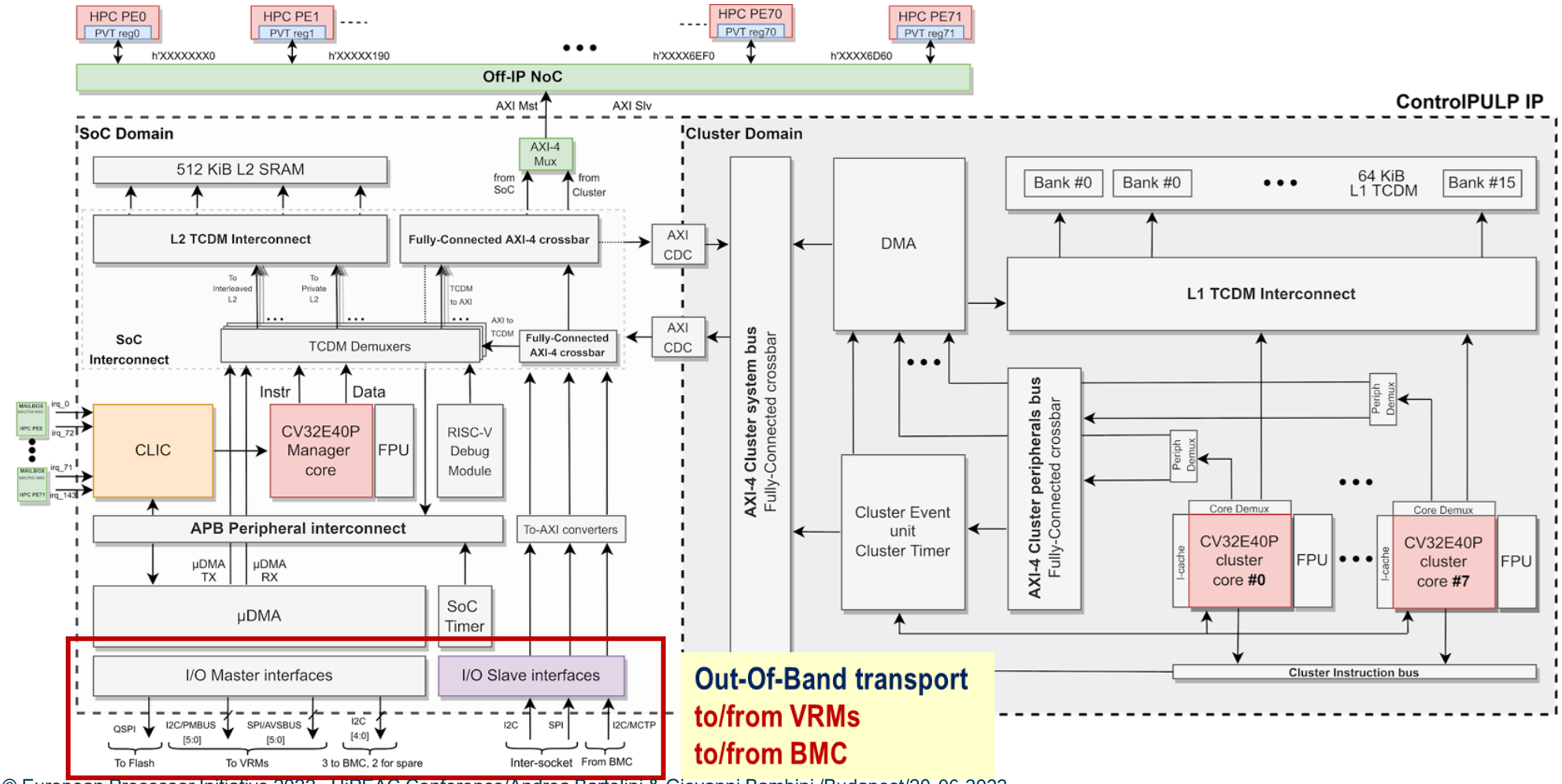
CONTROLPULP PROJECT: ARCHITECTURE



CONTROLPULP PROJECT: ARCHITECTURE

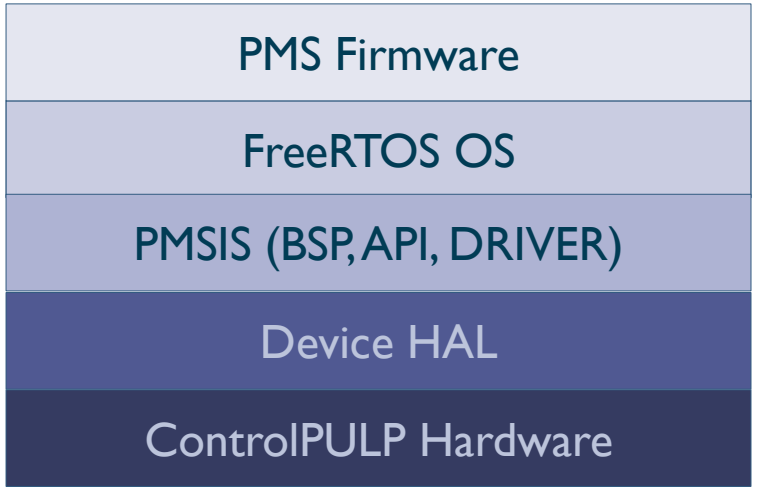


CONTROLPULP PROJECT: ARCHITECTURE



CONTROLPULP PROJECT: FIRMWARE

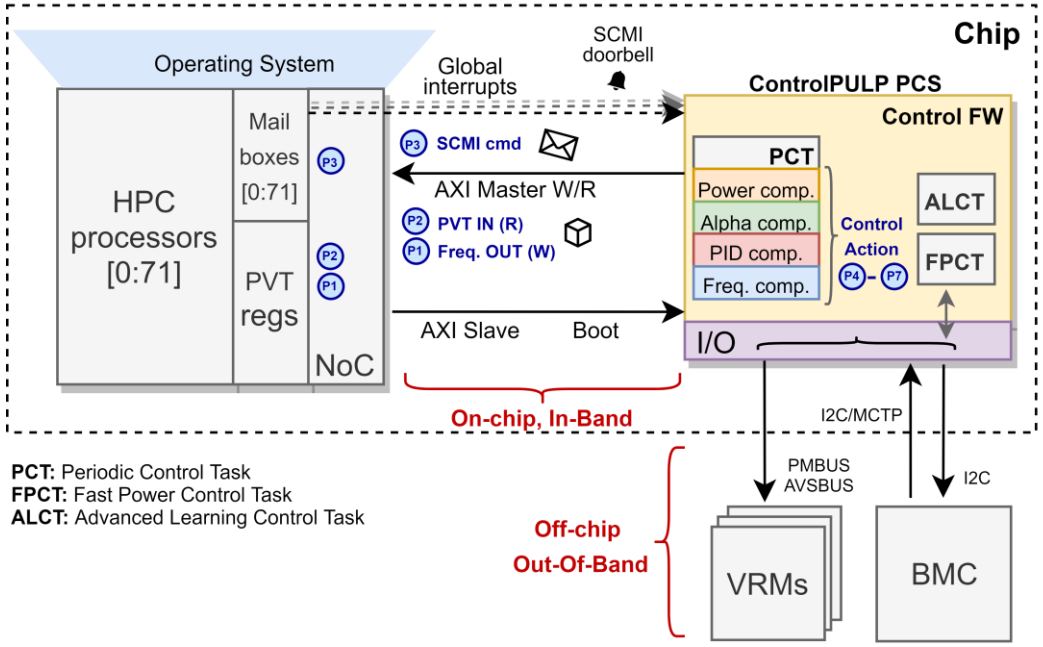
- Built on top of FreeRTOS (which is open-source)
  - Real Time scheduler with Pre-emption and priority-based task selection



[2] G. Bambini et al., "An Open-Source Scalable Thermal and Power Controller for HPC Processors", 2020

CONTROLPULP PROJECT: FIRMWARE

- Built on top of FreeRTOS (which is open-source)
  - Real Time scheduler with Pre-emption and priority-based task selection
- Three main control tasks[2]:
  1. Fast Power Control Task (FPCT) – 125 us
  2. Periodic Control Task (PCT) – 500us
  3. Advanced Control Task (ALCT) – 2000us
- Built with Modularity and hardware flexibility in mind

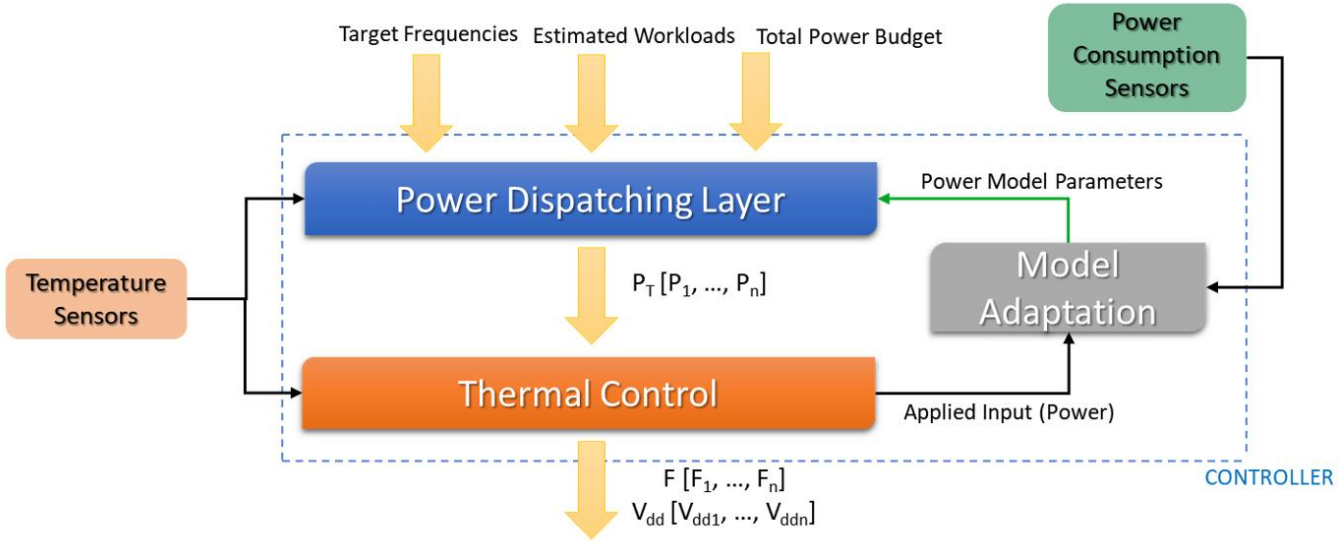


[2] G. Bambini et al., "An Open-Source Scalable Thermal and Power Controller for HPC Processors", 2020

CONTROLPULP PROJECT: FIRMWARE

Control Algorithm:

- Power Dispatching Layer:
  - Computes the operating point to enforce the given power budget.
  - It privileges more demanding cores.
  - Control performance is improved by model adaptation and/or ML algorithms



- Thermal Control:
  - Distributed algorithm which generates a power management setting for each tile which full-fills the allocated power budget and thermal constraints
  - Based on a PID and a power model inversion
- Frequency Binding:
  - If enforced, will apply the same DVFS operating point to the core executing the same application

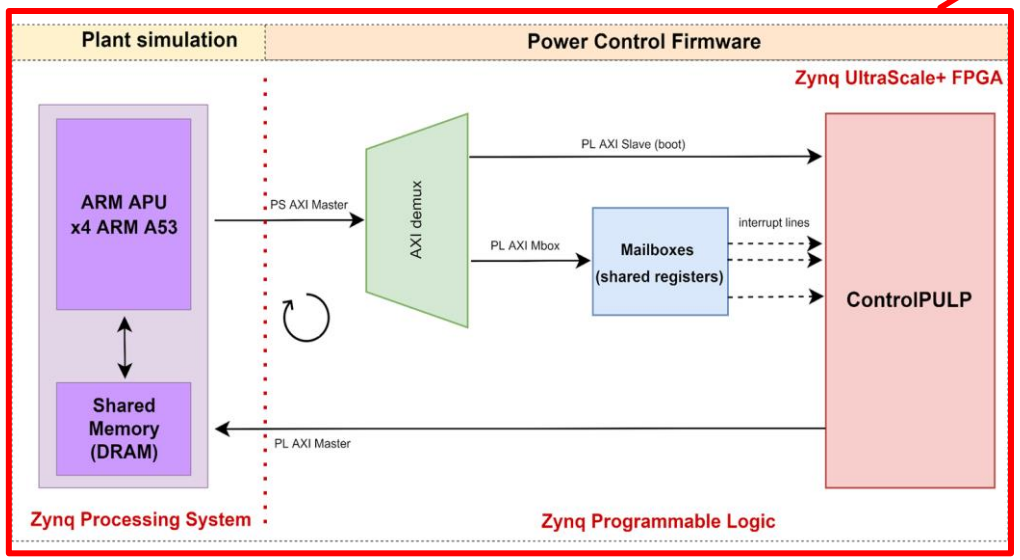
OUTLINE

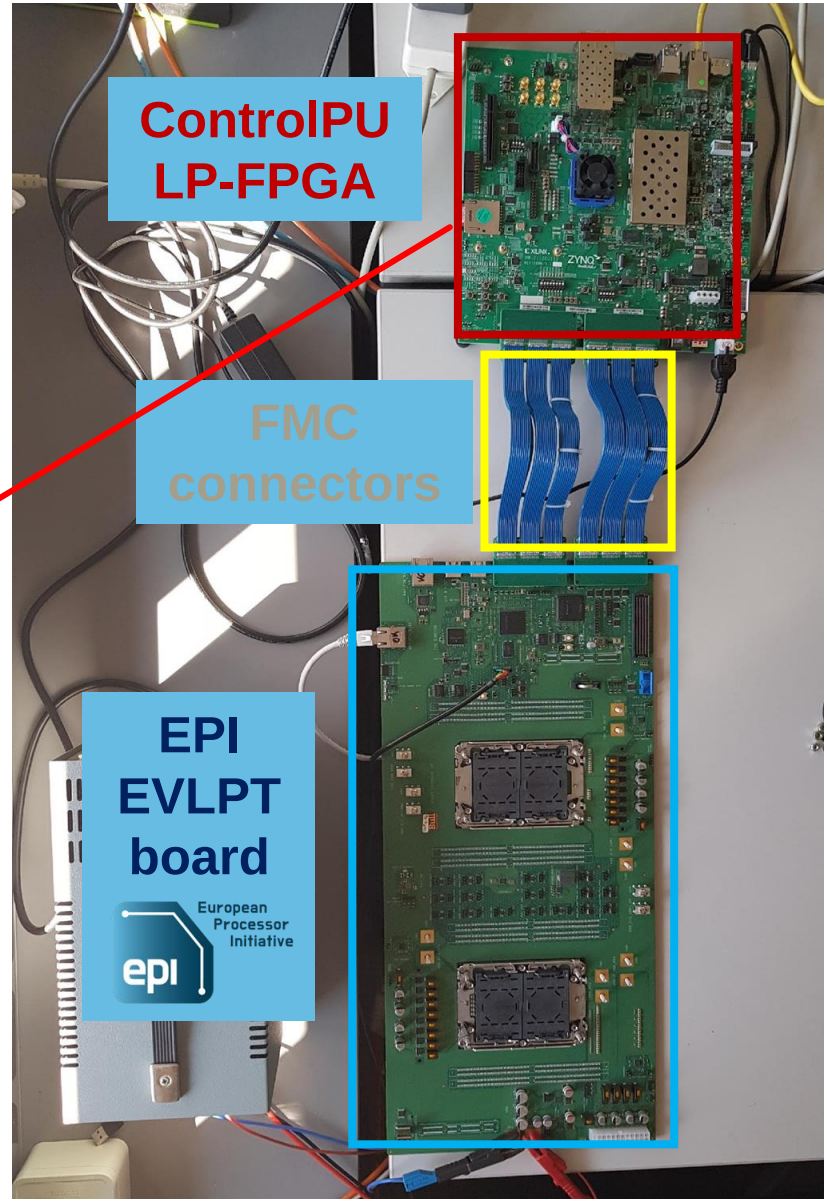
- Power Management in HPC
- ControlPulp Project: an open-source hardware/software RISC-V controller
- **ControlPULP HIL Co-Desing Framework**
- Integration and Demo
- QnA

CONTROL PULP CODESIGN

FPGA-based Hardware-in-the-Loop emulation

- RTL + FW @ FPGA

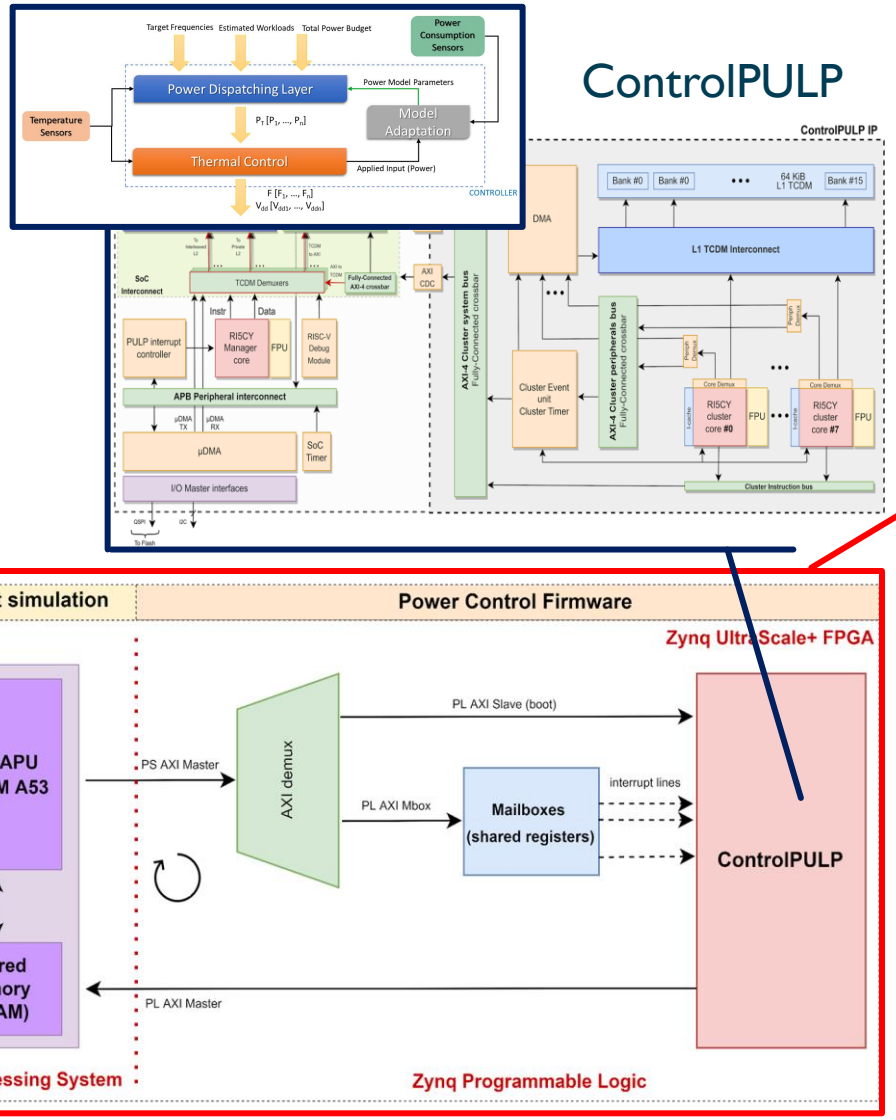


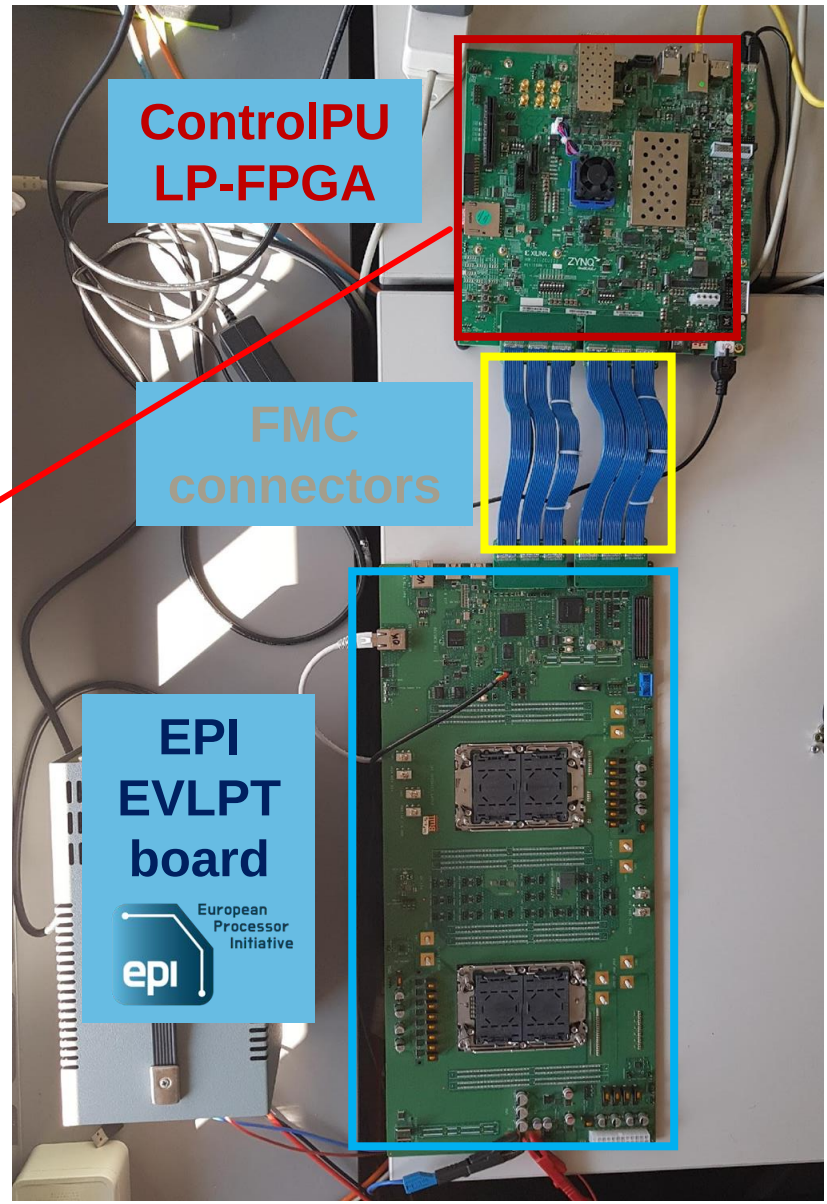


CONTROL PULP CODESIGN

FPGA-based Hardware-in-the-Loop emulation

- RTL + FW @ FPGA

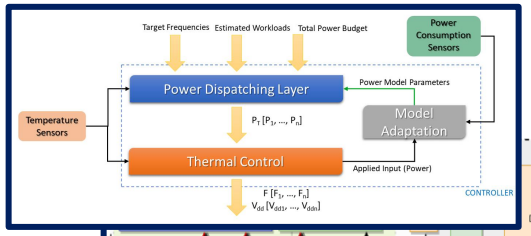




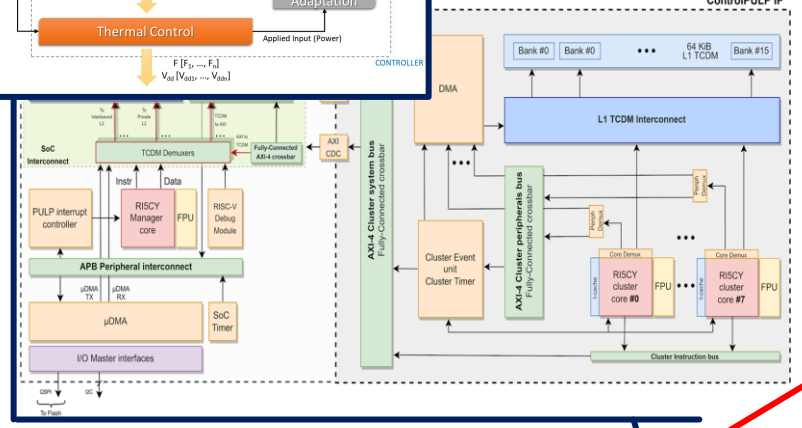
CONTROL PULP CODESIGN

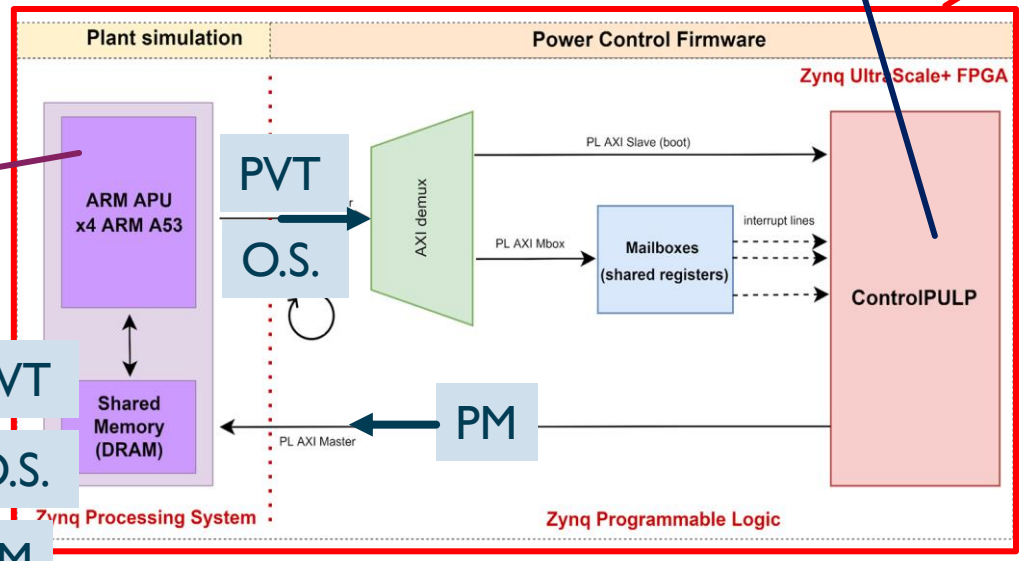
FPGA-based Hardware-in-the-Loop emulation

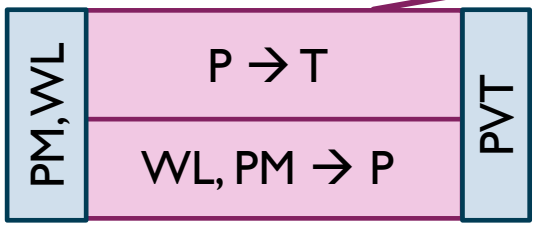
- RTL + FW @ FPGA
- PLANT sim. @ A53



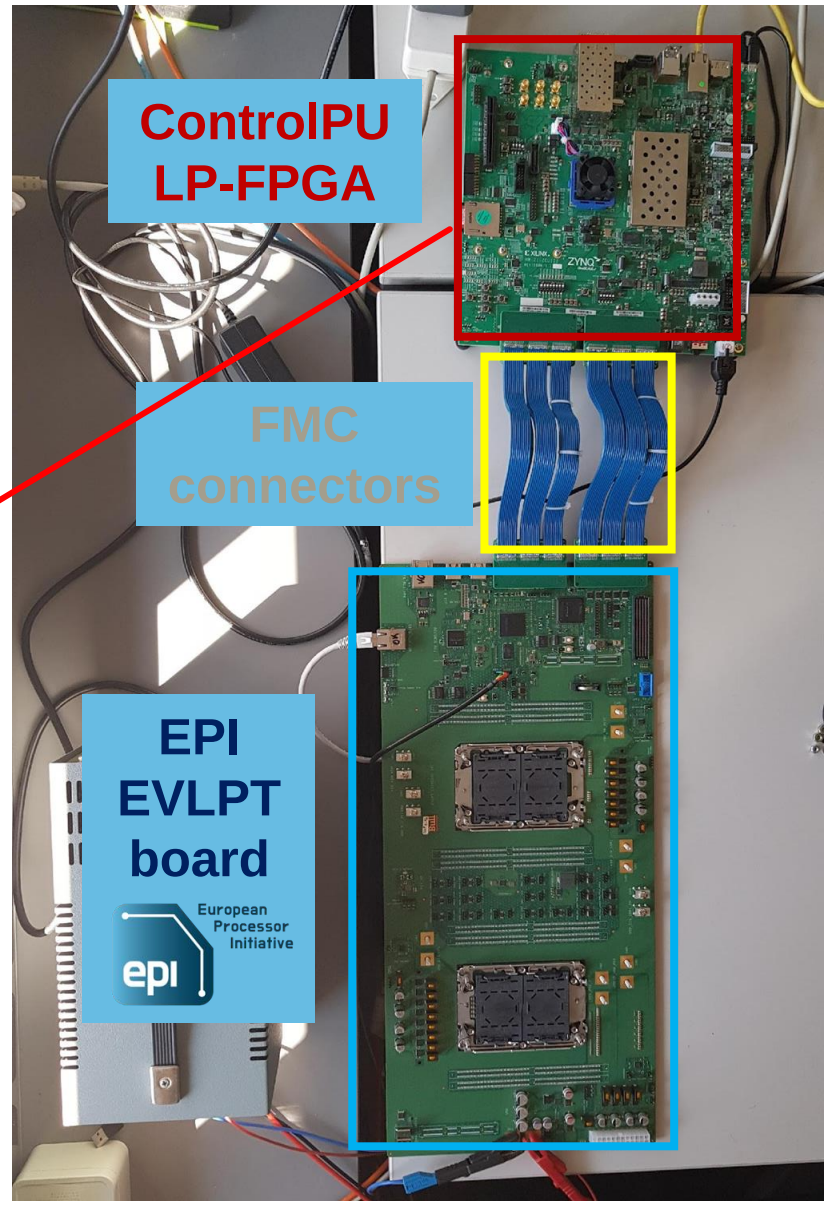
ControlPULP







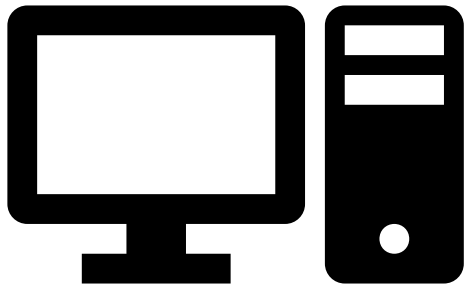
Perf./Power/Thermal Models

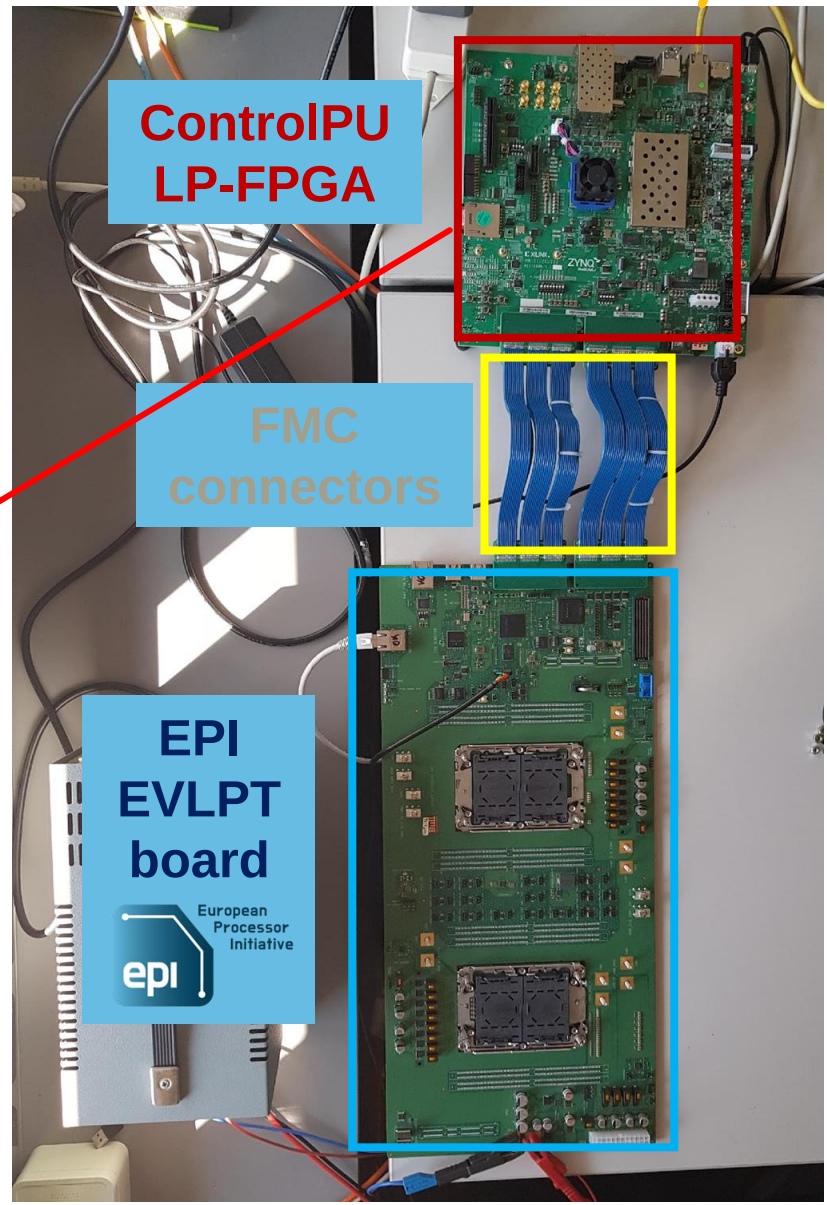


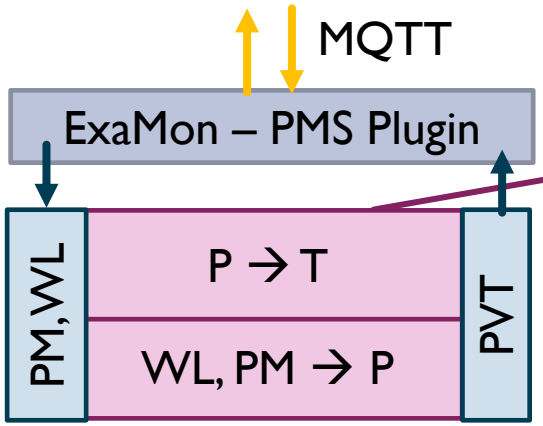
CONTROL PULP CODESIGN

FPGA-based Hardware-in-the-Loop emulation

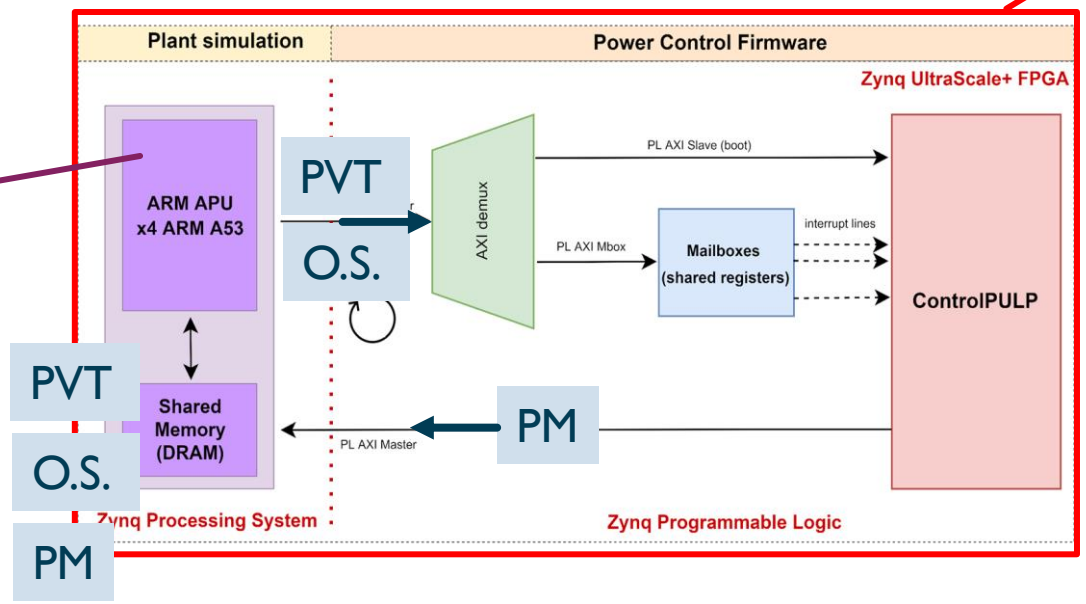
- RTL + FW @ FPGA
- PLANT sim. @ A53
- ExaMon integration







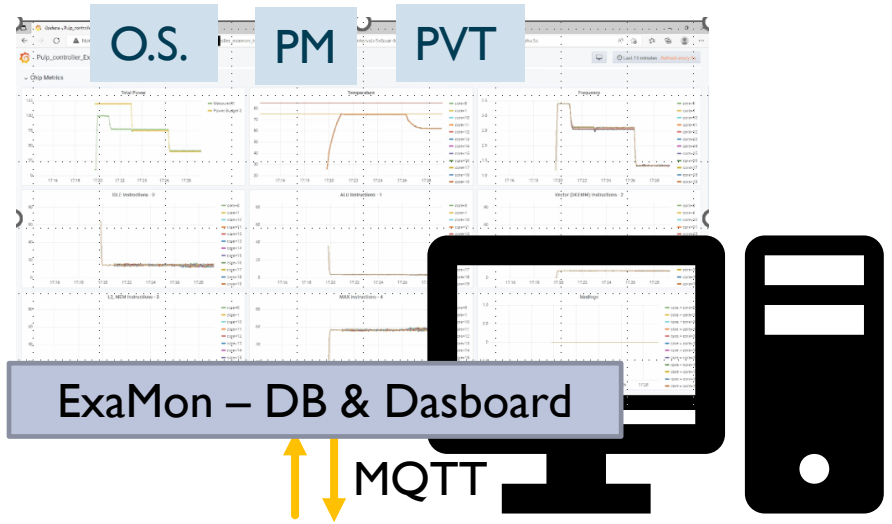
Perf./Power/Thermal Models

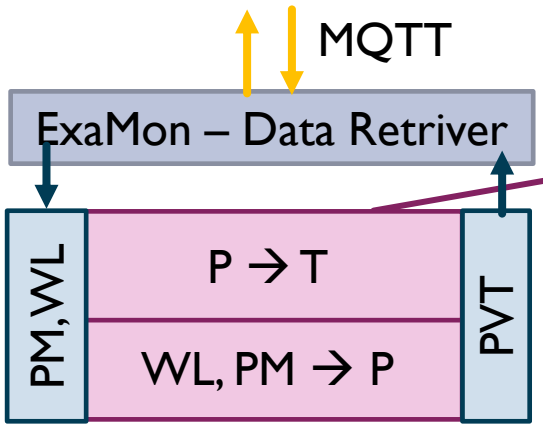


CONTROL PULP CODESIGN

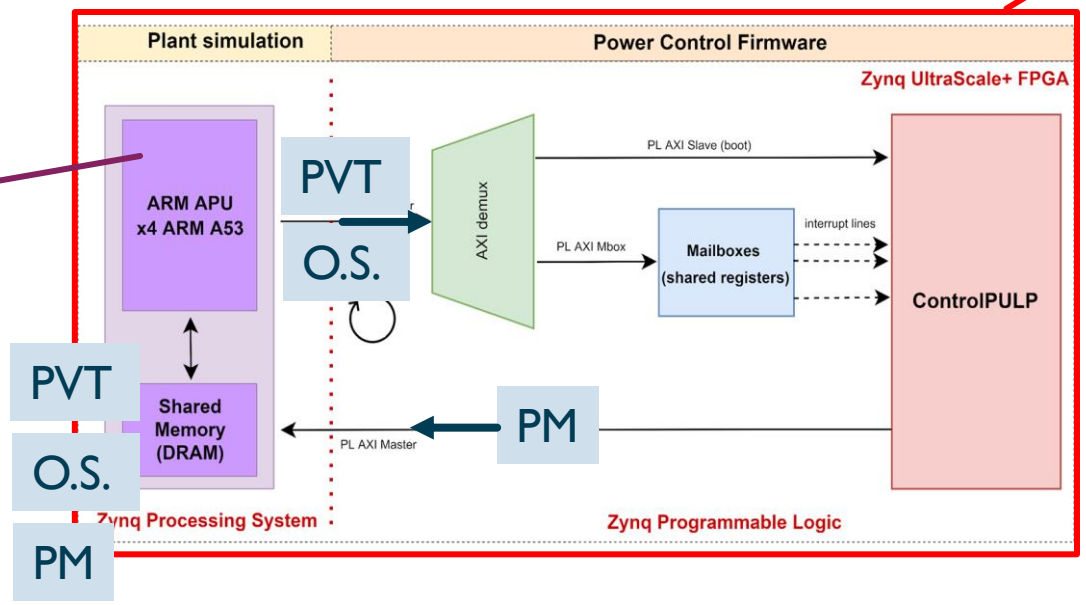
FPGA-based Hardware-in-the-Loop emulation

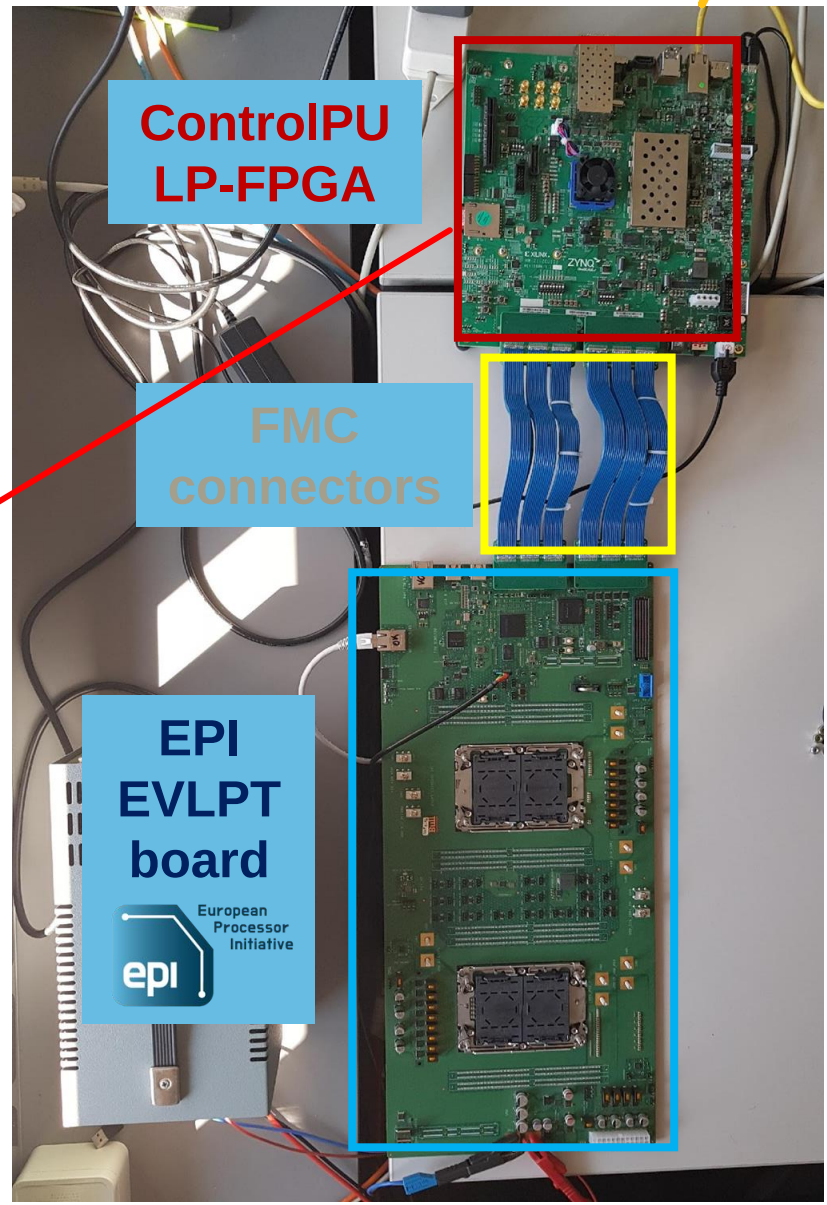
- RTL + FW @ FPGA
- PLANT sim. @ A53
- ExaMon integration





Perf./Power/Thermal Models



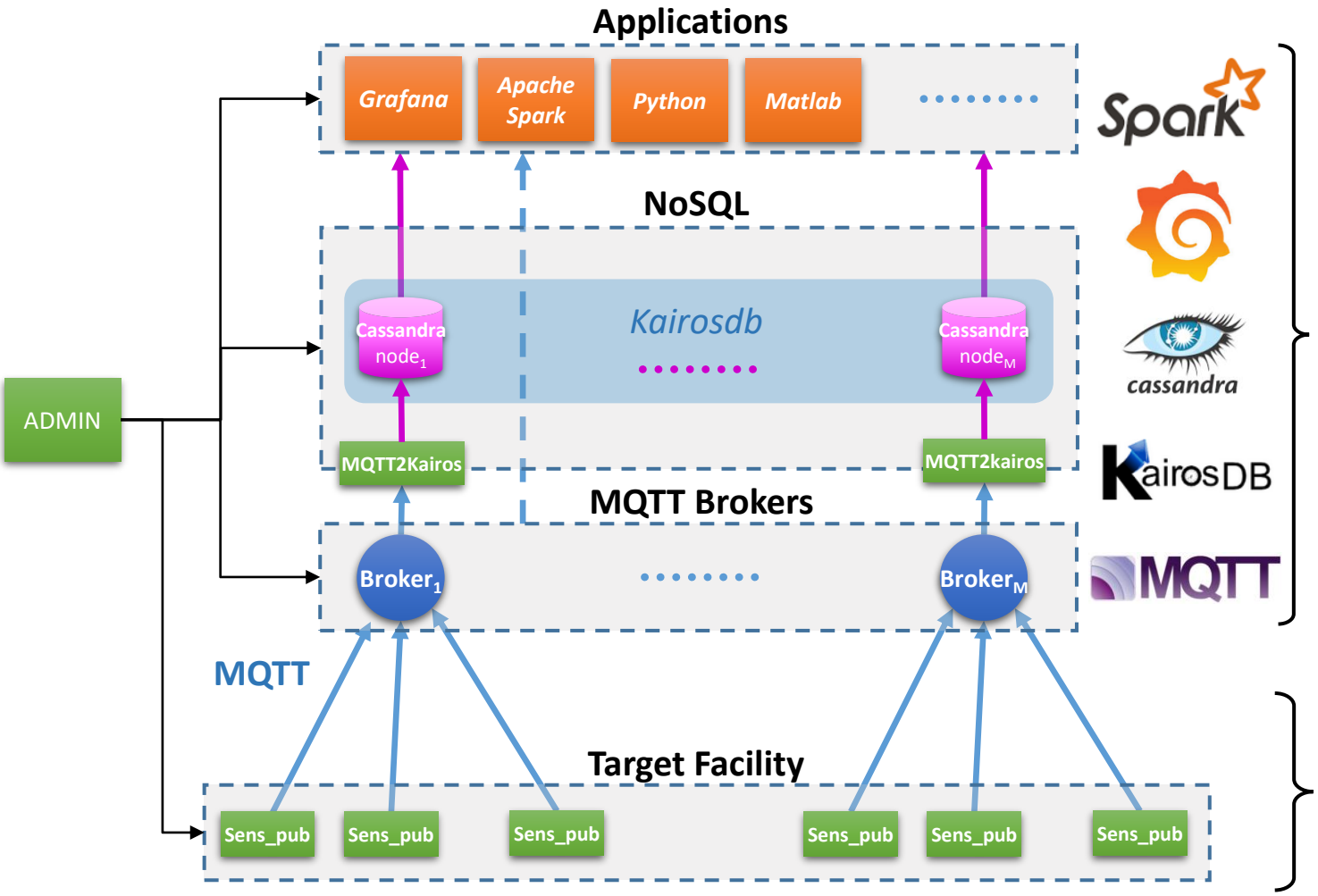


```
-bash-4.2$
```

WHY EXAMON

- ExaMon was primarily developed, in collaboration with **CINECA**, to collect and analyze HPC nodes and facility data.
- Early 2015
  - Galileo 2015
  - Marconi 2016
  - D.A.V.I.D.E. 2017-2020
  - Marconi100 2020
  - Galileo 100 2021
- **Current deploy:** Marconi (SKL), Marconi100, Galileo100
  - Nodes: ~6100
  - DB size: ~28TB (on-line)
  - >1Million unique sensors

WHY EXAMON



Front-end

- **MQTT** Brokers
- Data **Visualization**
- **NoSQL** Storage
- **Big Data** Analytics

Back-end

- MQTT–enabled sensor collectors

ExaMon (Exascale Monitoring) is a data collection and analysis framework oriented to the management of big data

OUTLINE

- Power Management in HPC
- ControlPulp Project: an open-source hardware/software RISC-V controller
- ControlPULP HIL Co-Desing Framework
- **Co-simulation Demo**
- QnA

CONTROL PULP - DEMO

SCALABLE RISC-V POWER CONTROLLER PLATFORM FOR HPC PROCESSORS

Andrea Bartolini, **Giovanni Bambini**, University of Bologna

CO-SIMULATION DEMO: INTRO

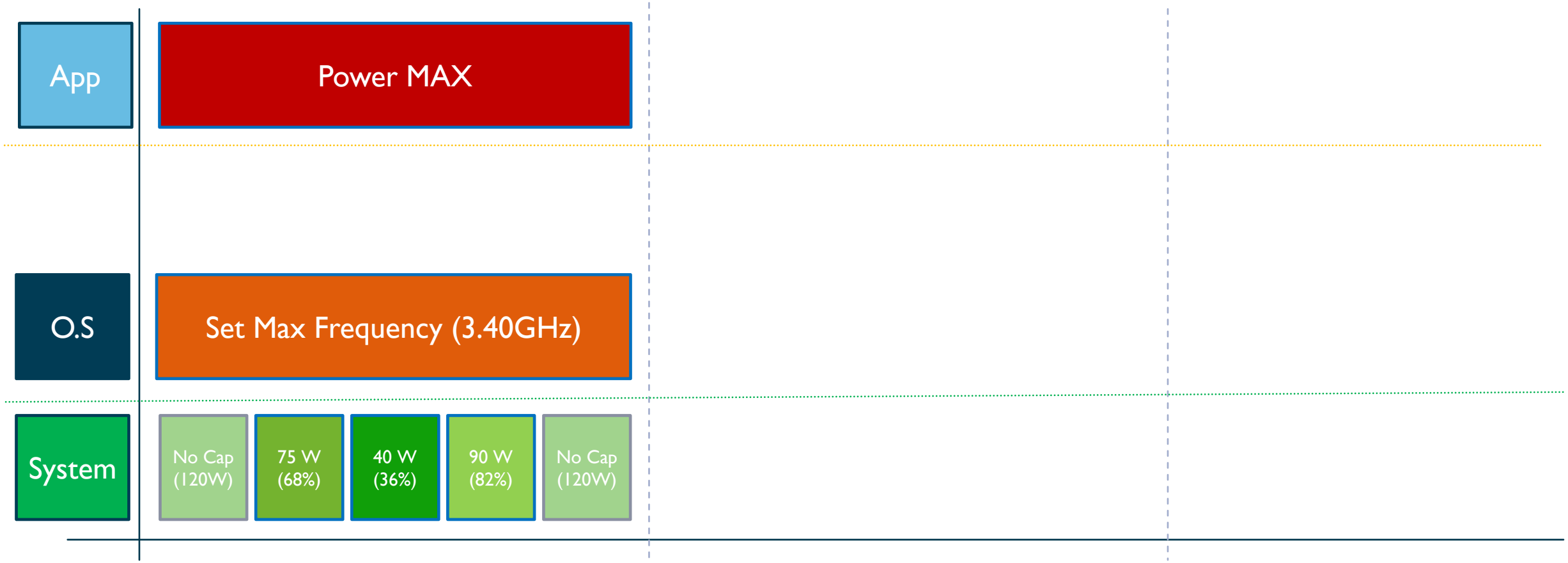
- The demo is structured in 3 Phases:
  1. Focus on the Power Capping and Thermal Capping capabilities
  2. Focus on the workload adaptation capabilities of the control policy
  3. Focus on the O.S. interaction and frequency set point tracking
- Co-simulation settings:
  - The simulation executes at 1/200x of Real Time
  - EXAMON data are gathered with a sampling time of **1.25ms** (simulated time)
  - We used synthetic workload composed by corner cases power characteristics

CO-SIMULATION DEMO: 1° PHASE

- Phase 1°: Power Max (~ 5s)
  - The instructions executed make the **maximum power consumption** possible: all core components are active (Vector and Alu operations included).
  - There are **5 subphases** in which the power capping commands is changed:
    - A) No capping, 1s
      - this subphase shows the **thermal capping**
    - B) 75W (68%) capping, 1s
    - C) 40W (36%) capping, 1s
    - D) 90W (82%) capping, 1s
    - E) No capping, ~1s

** Each time is expressed in Simulated Time*

CO-SIMULATION DEMO: 1° PHASE

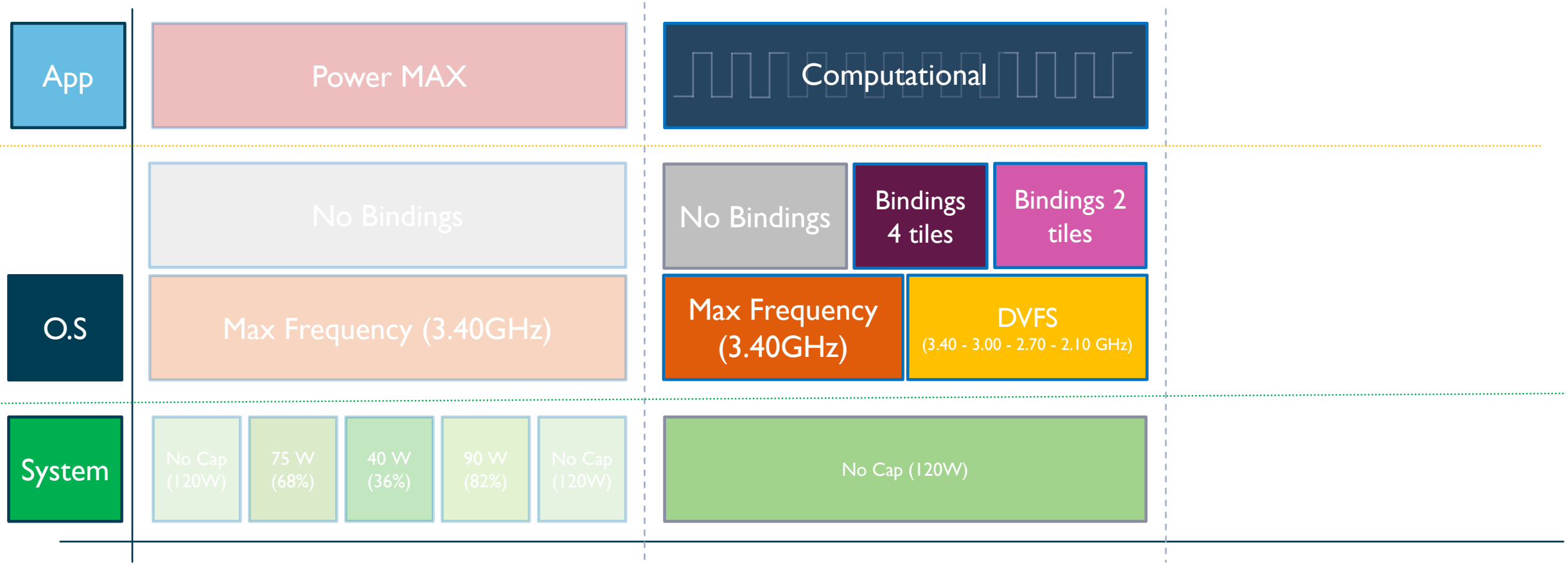


CO-SIMULATION DEMO: 2° & 3° PHASES

- Phase 2°: Computation (~ 5s)
  - There are 16 **computation subphases** (combination of **Vector** (~ 30%) and **L2_MEM** (~ 50%) and **ALU** (~ 15%) instructions) alternated by 8 **waiting subphases**.
- Phase 3°: DVFS (~ 8s)
  - «Random» **DVFS** commands sent to each core.
  - Instructions executed are a combination of **IDLE, ALU, Vector (DGEMM),** and **L2_MEM**.
- Common to both phases: Bindings (~ 6s)
  - It starts during the Computation Phase
  - With Bindings we *force* **binded core groups** to run at the same frequency: in this way, we can save energy (or allocate it to other cores) while executing parallel code that is expected to synchronously execute on all cores in the group

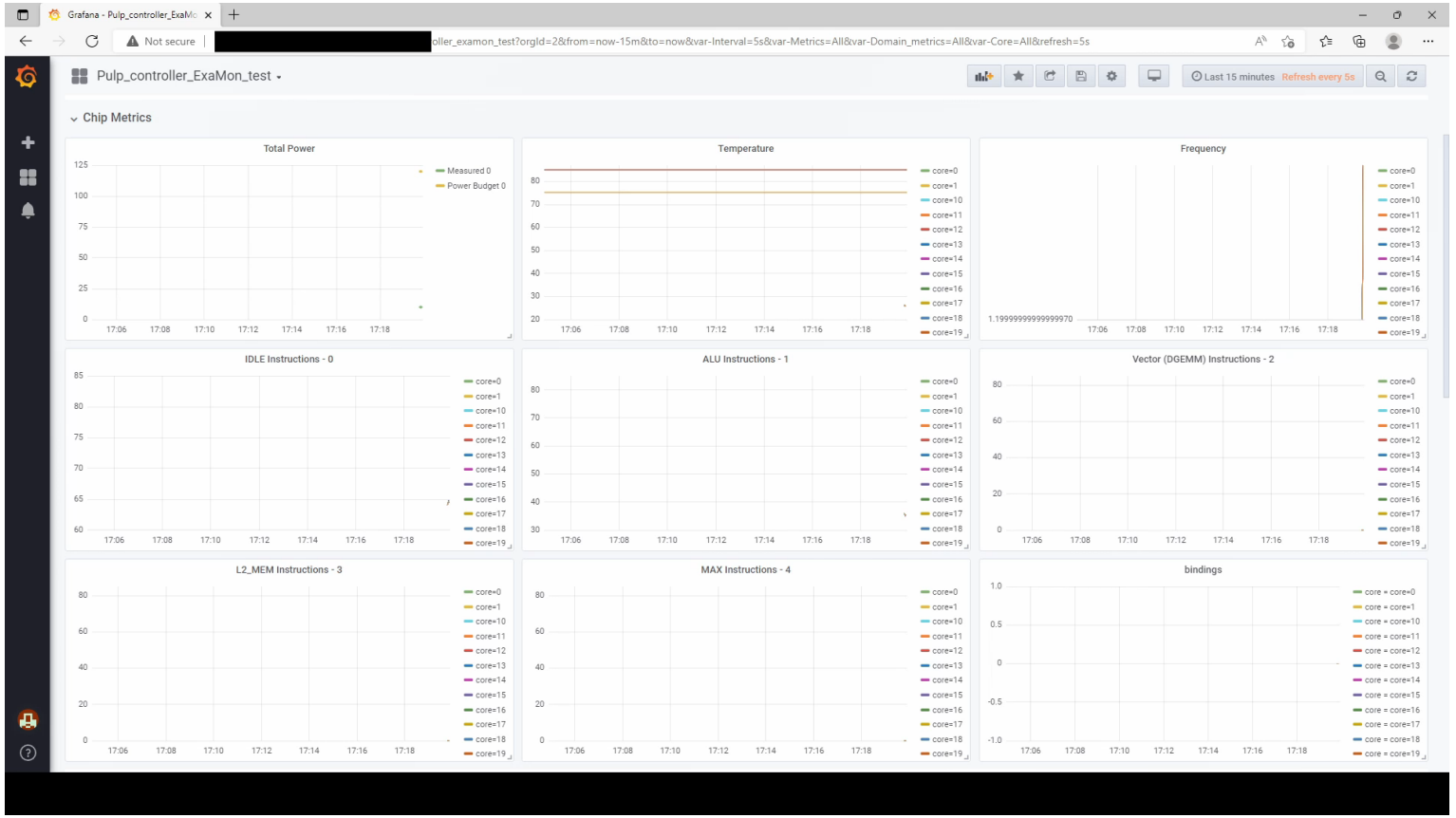
** Each time is expressed in Simulated Time*

CO-SIMULATION DEMO: 2° PHASE

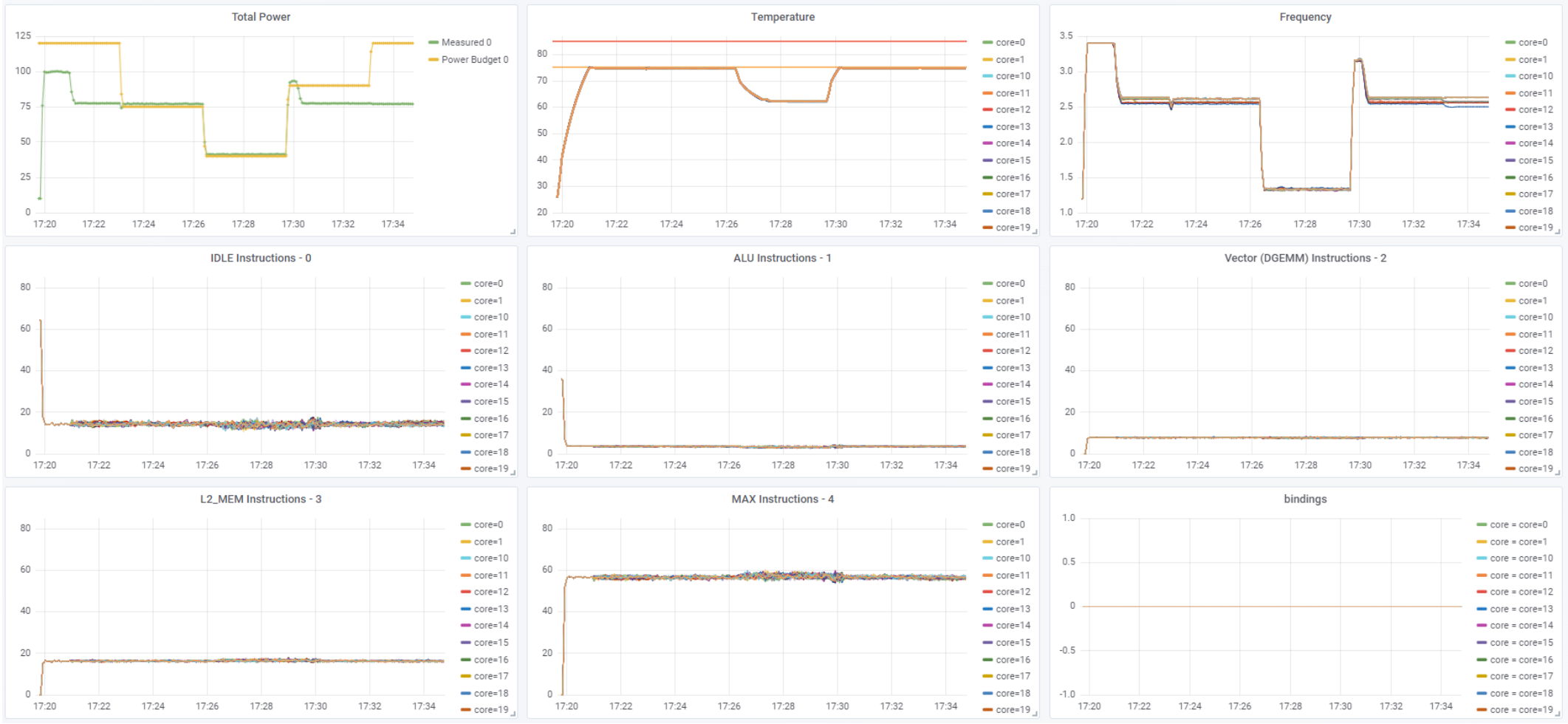


CO-SIMULATION DEMO: 3° PHASE

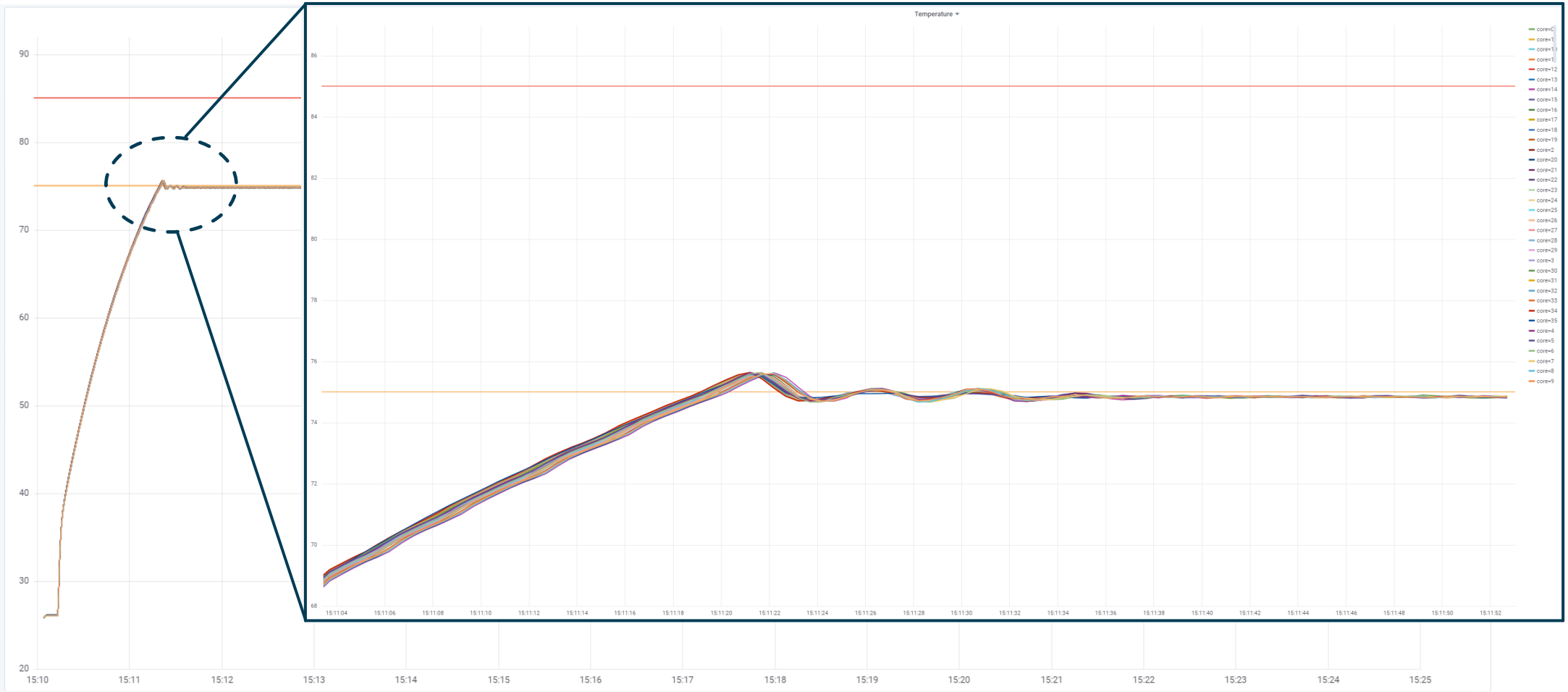
CO-SIMULATION DEMO: 1° PHASE



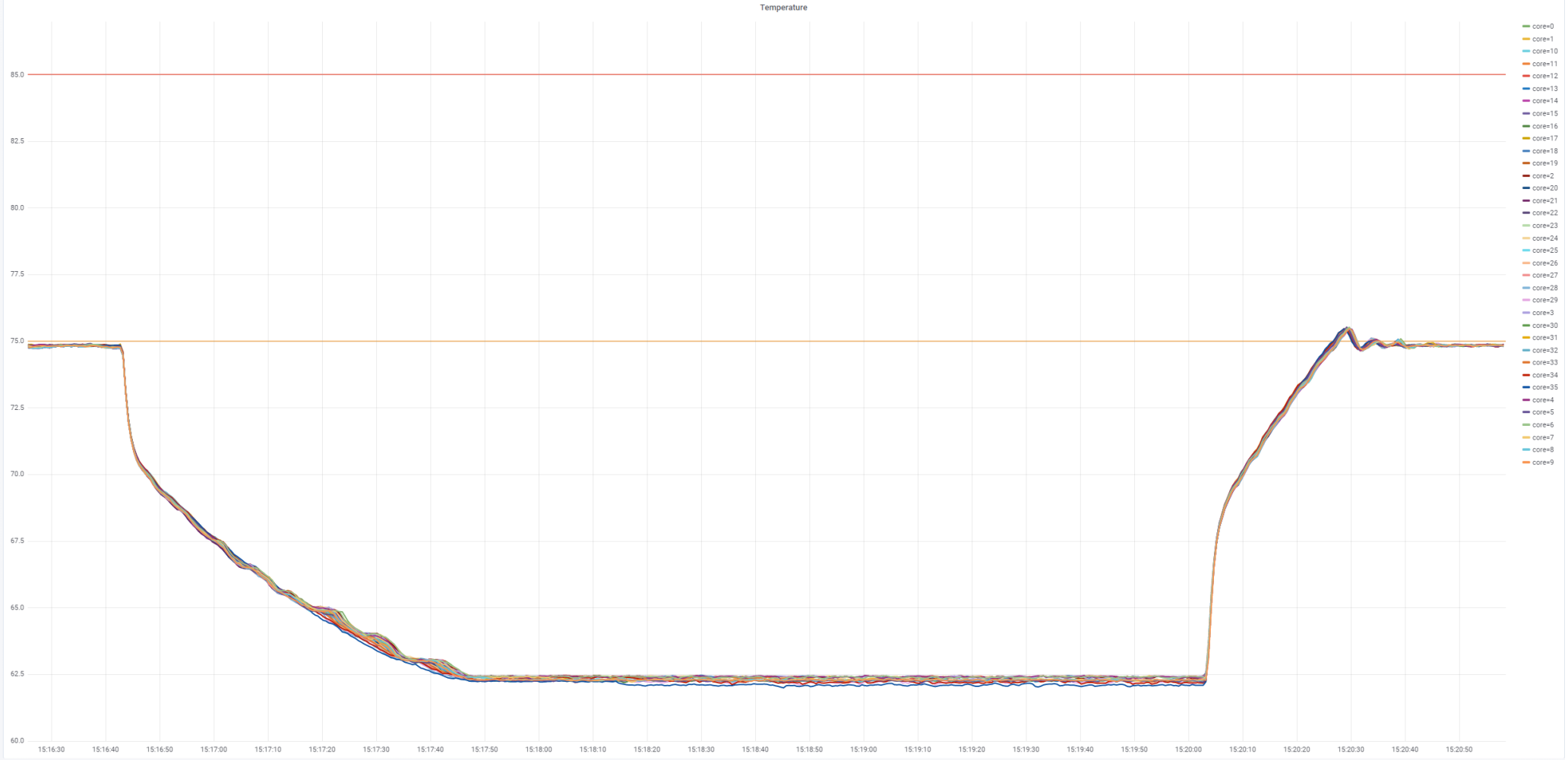
CO-SIMULATION DEMO: 1° PHASE



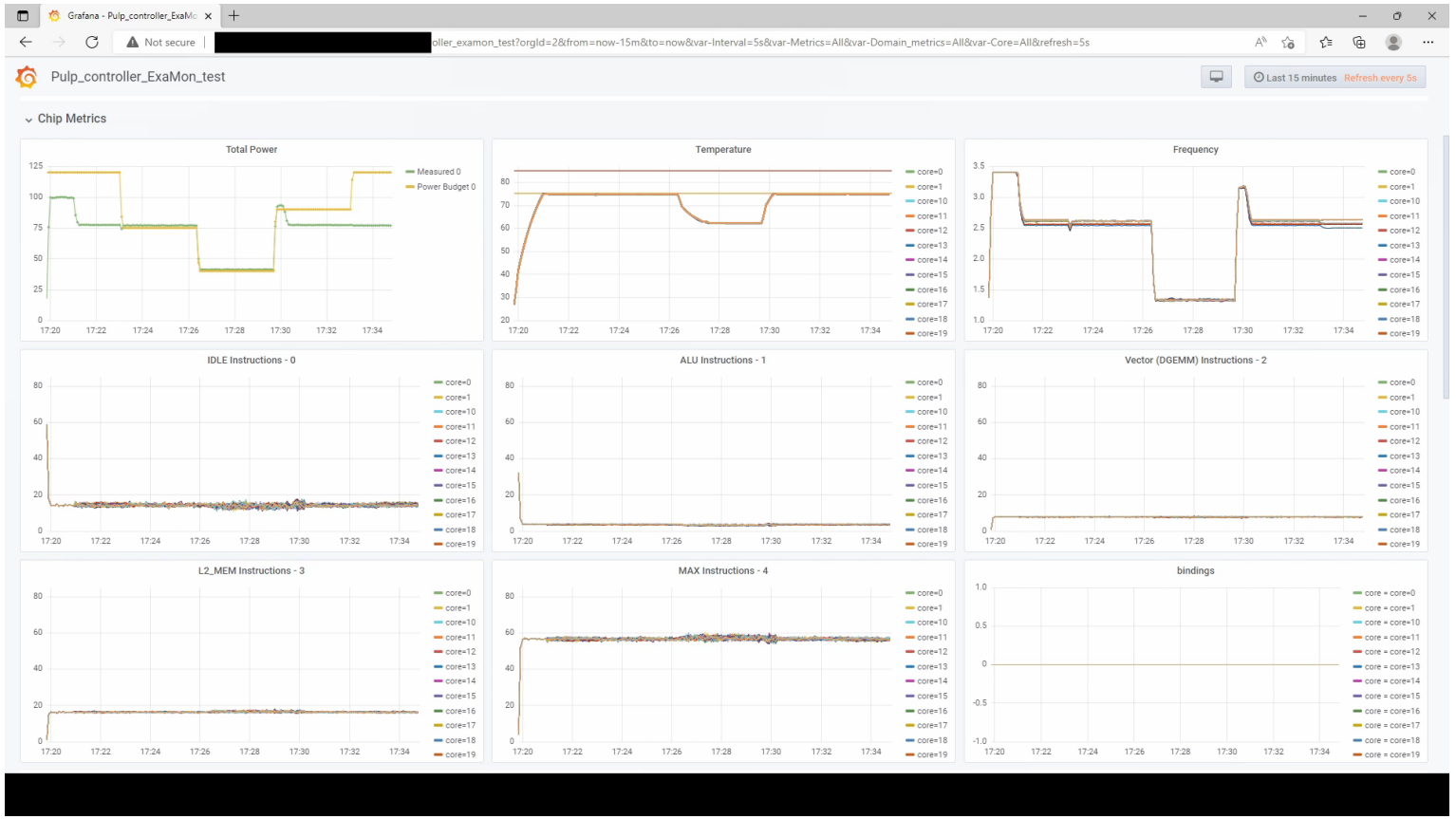
CO-SIMULATION DEMO: 1° PHASE THERMAL CAPPING



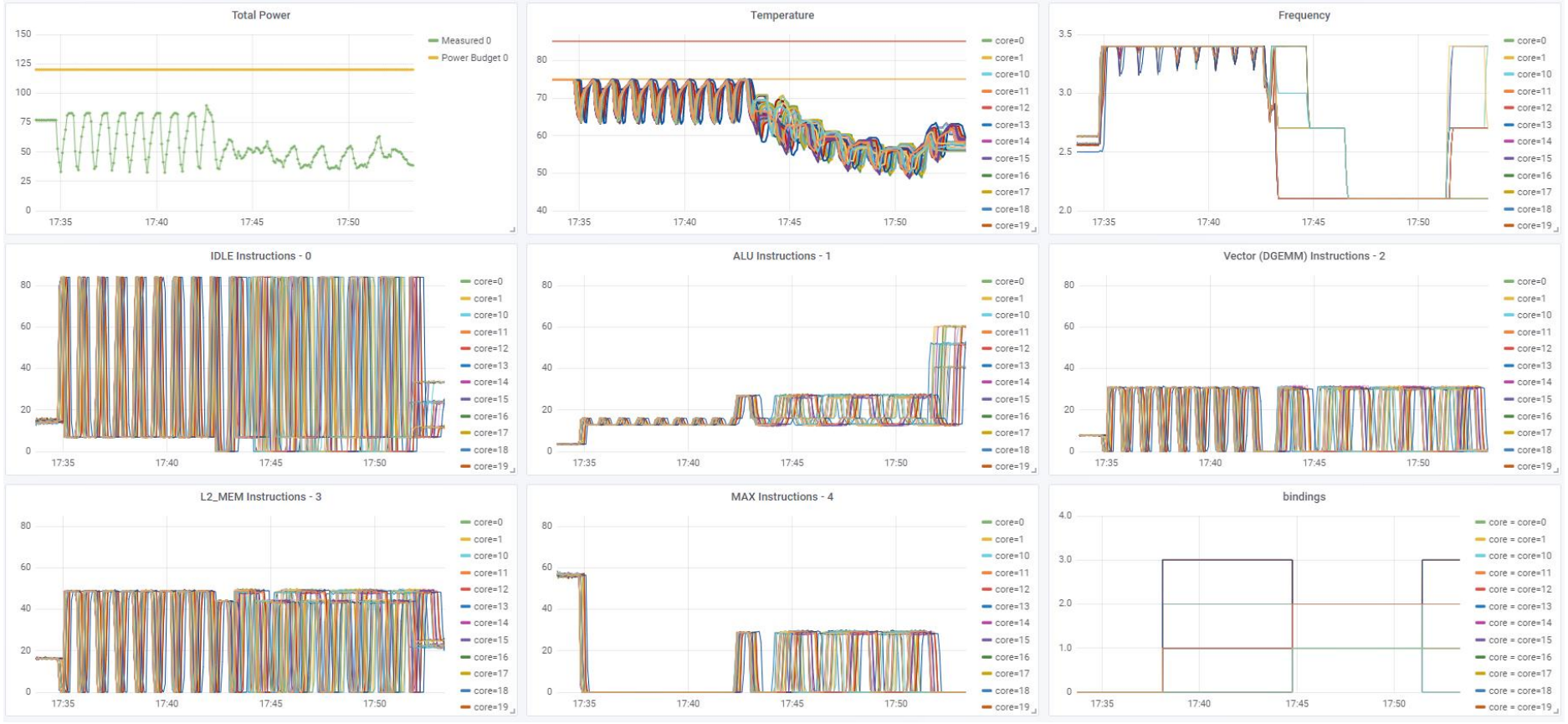
CO-SIMULATION DEMO: 1° PHASE THERMAL CAPPING



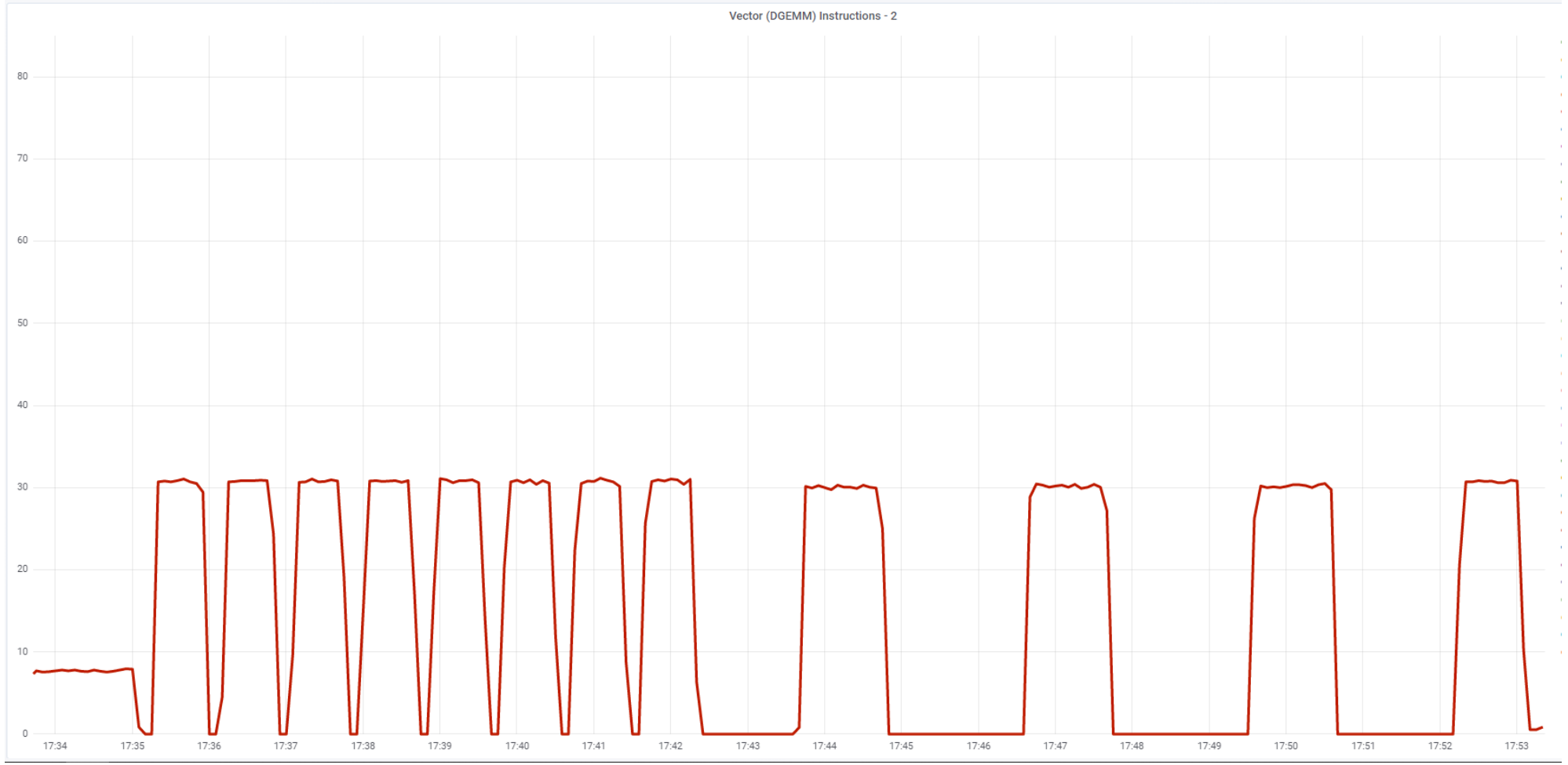
CO-SIMULATION DEMO: 2° PHASE



CO-SIMULATION DEMO: 2° PHASE



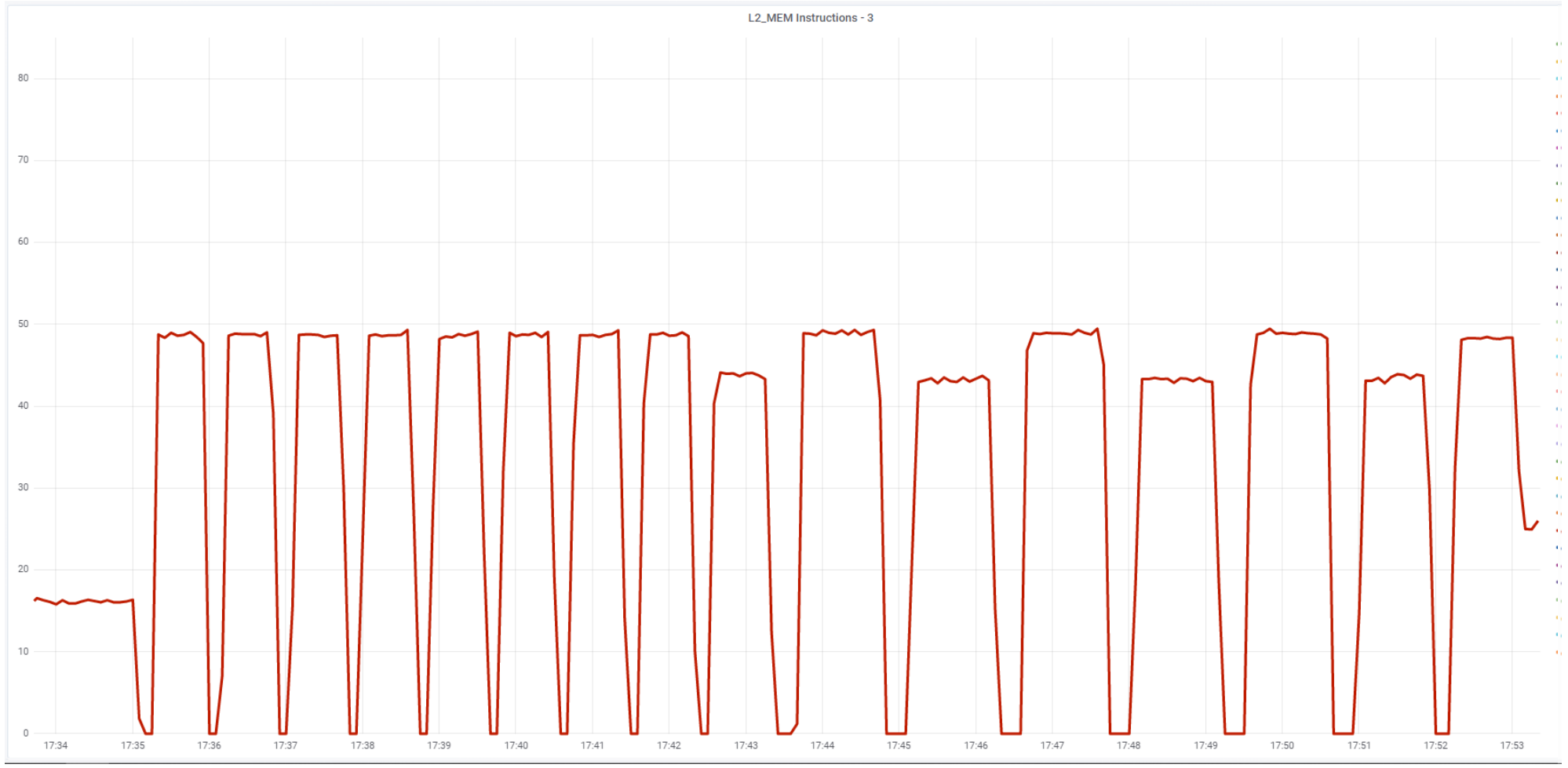
CO-SIMULATION DEMO: 2° PHASE WORKLOAD COMPOSITION



IDLE Instructions

Vector Instructions

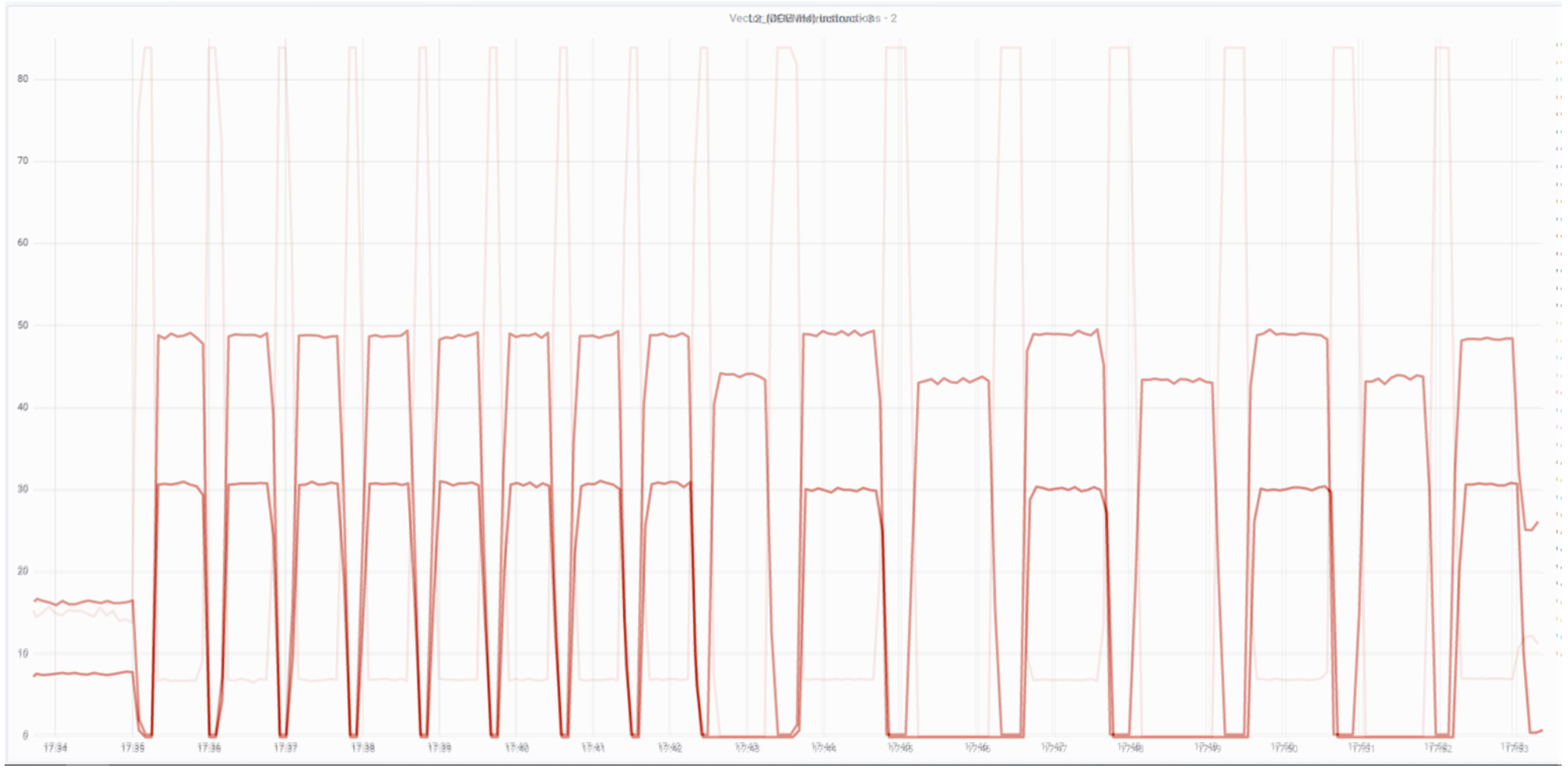
CO-SIMULATION DEMO: 2° PHASE WORKLOAD COMPOSITION



Vector Instructions

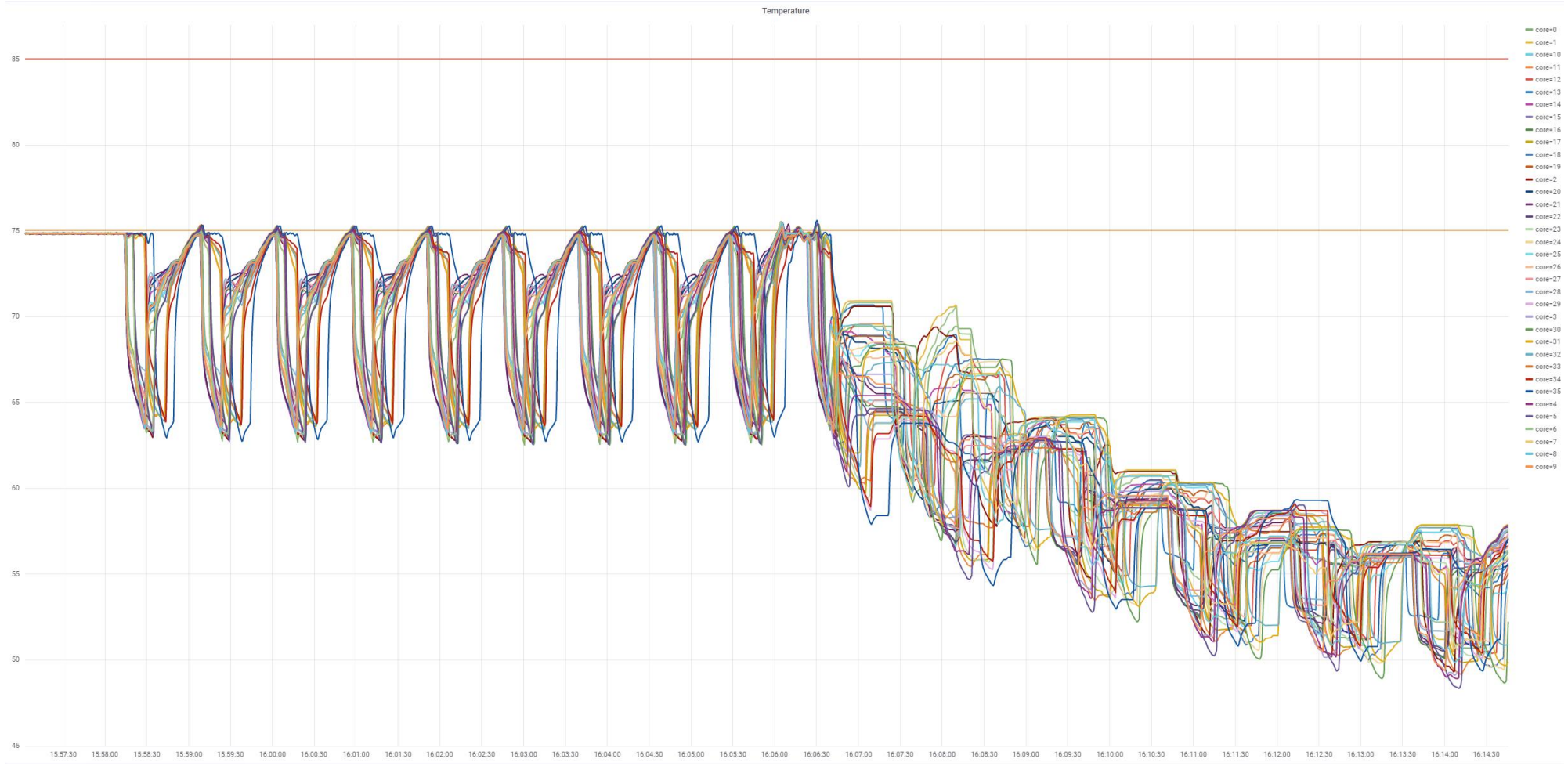
L2_MEM Instructions

CO-SIMULATION DEMO: 2° PHASE WORKLOAD COMPOSITION



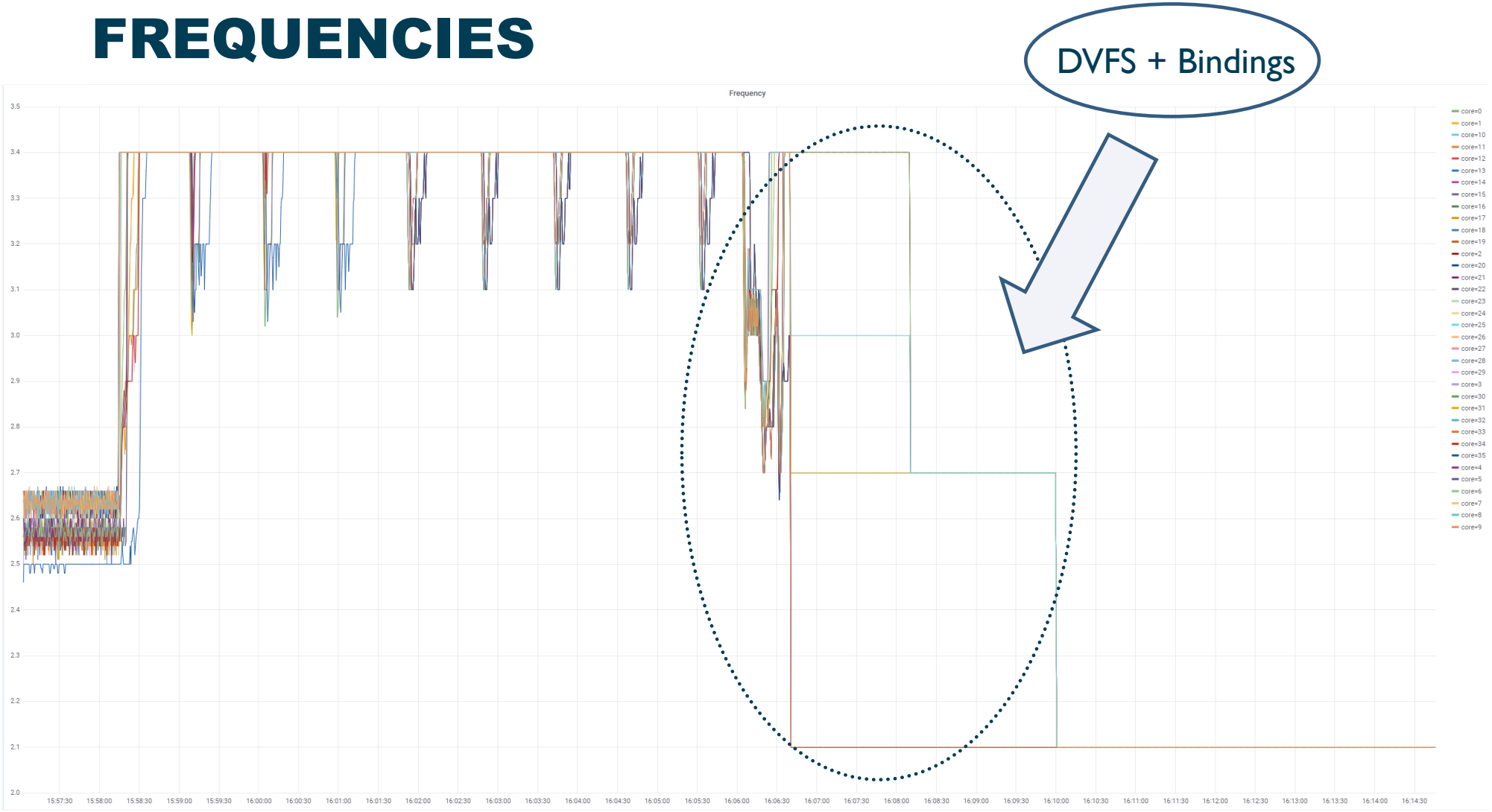
L2_MEM Instructions

CO-SIMULATION DEMO: 2° PHASE TEMPERATURE



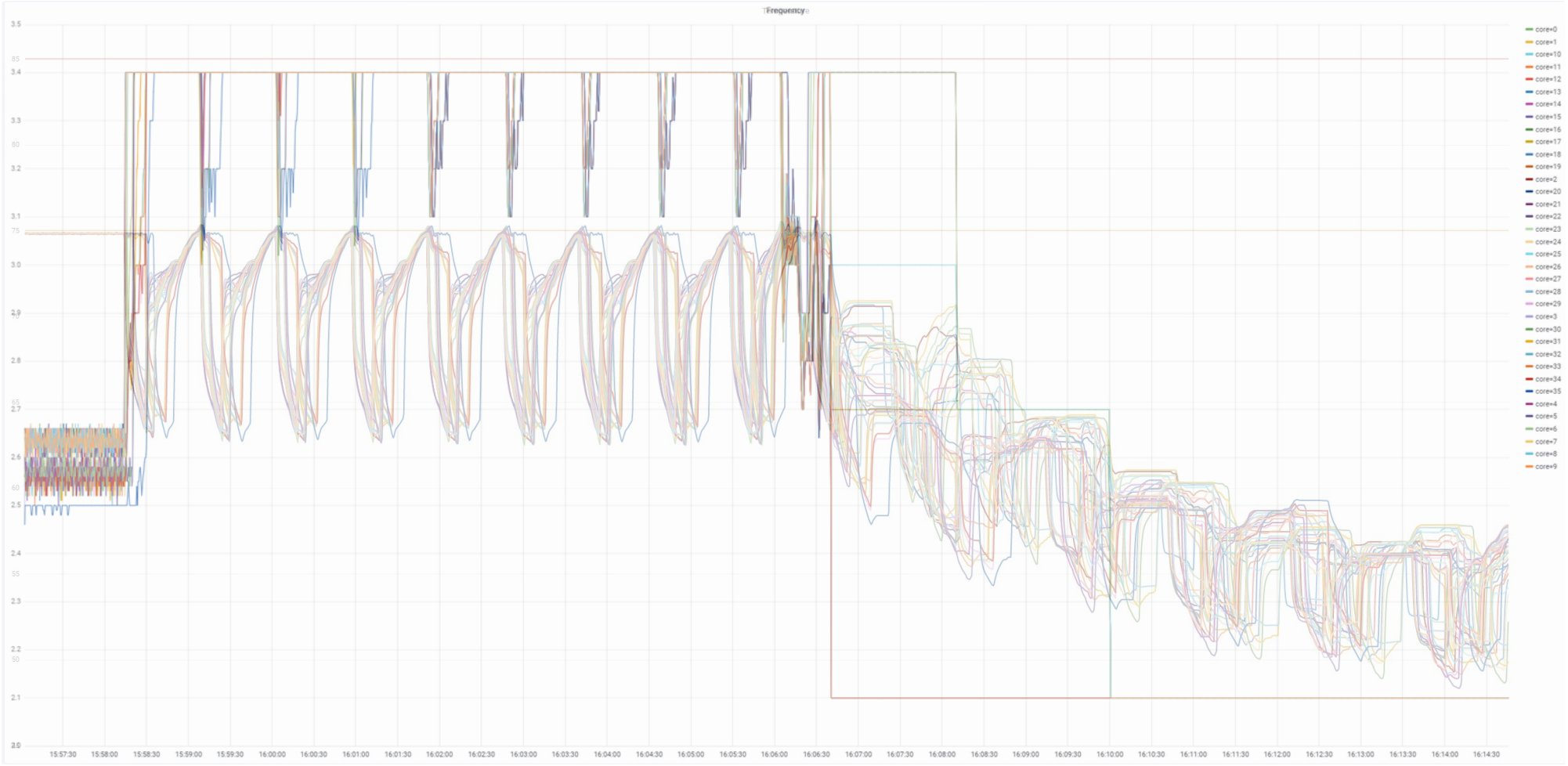
Temperature

CO-SIMULATION DEMO: 2° PHASE FREQUENCIES



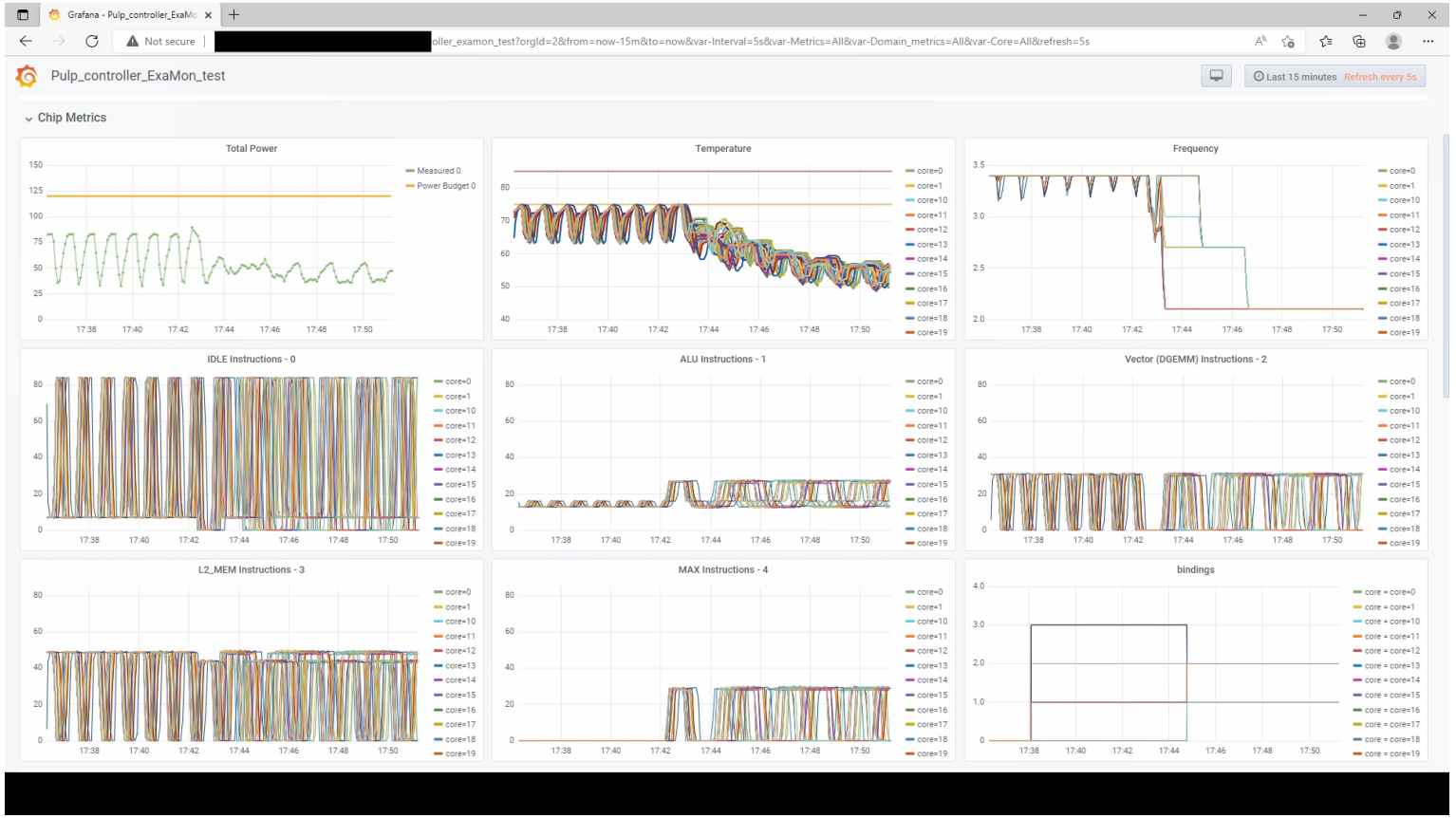
Frequency

CO-SIMULATION DEMO: 2° PHASE FREQUENCIES

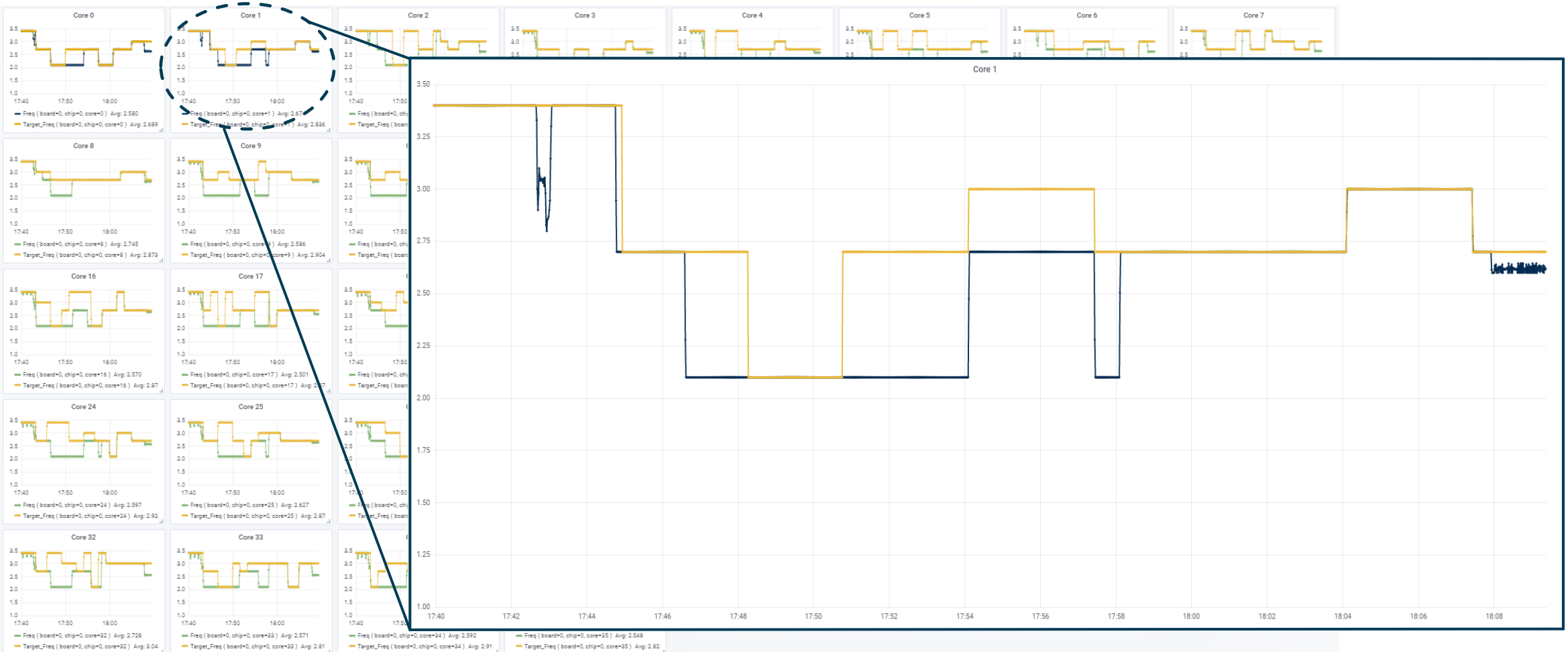


Frequency

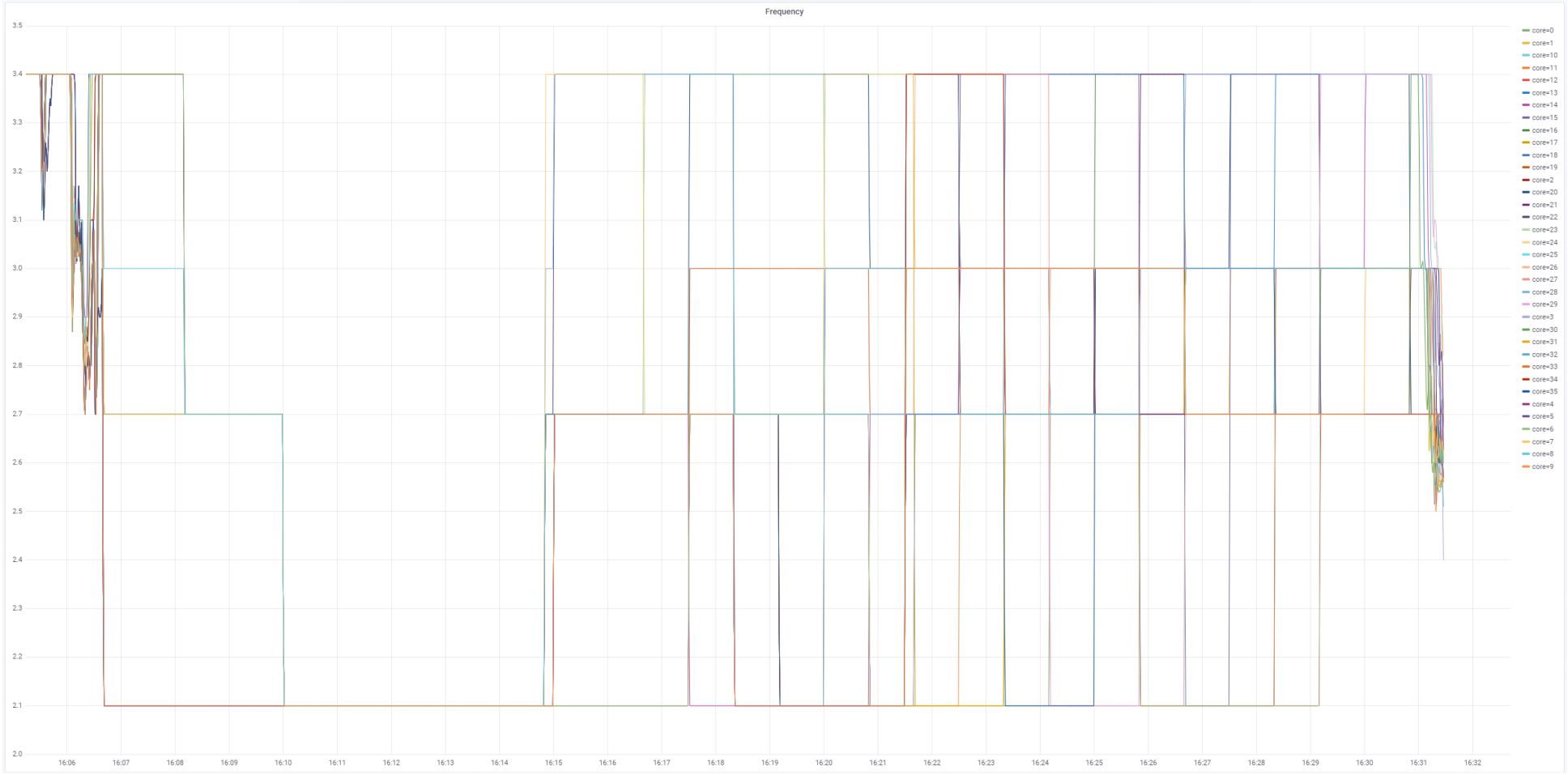
CO-SIMULATION DEMO: 3° PHASE



CO-SIMULATION DEMO: 3° PHASE



CO-SIMULATION DEMO: 3° PHASE FREQUENCIES + BINDINGS



CONCLUSION

- Power control of High Performance Computing processor is a demanding task
- ControlPULP is a flexible and powerfull open-source controller which can be integrated in HPC processors
- We designed a HIL Co-design framework for the design, test and validation of ControlPULP.
- The HIL Co-design framework will be released w. ControlPULP.

www.european-processor-initiative.eu

@EuProcessor

European Processor Initiative

European Processor Initiative

EPI FUNDING

This project has received funding from the European High Performance Computing Joint Undertaking (JU) under Framework Partnership Agreement No 800928 and Specific Grant Agreement No 101036168 EPI-SGA2. The JU receives support from the European Union's Horizon 2020 research and innovation programme and from Croatia, France, Germany, Greece, Italy, Netherlands, Portugal, Spain, Sweden, and Switzerland.

EPI PARTNERS

ACKNOLEDGMENT

REGALE
Open Architecture for Exascale Supercomputers

The ControlPULP Design Team:

- Giovanni Bambini, Robert Balas, Corrado Bonfanti, Antonio Mastrandrea, Davide Rossi, Simone Benatti, Luca Benini, Andrea Bartolini

The european-project-initiative has received funding from the European High Performance Computing Joint Undertaking (JU) under Framework Partnership Agreement No 800928 and Specific Grant Agreement No 101036168 (EPI SGA2). The JU receives support from the European Union's Horizon 2020 research and innovation programme and from Croatia, France, Germany, Greece, Italy, Netherlands, Portugal, Spain, Sweden, and Switzerland.

The European PILOT project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No.101034126. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Italy, Switzerland, Germany, France, Greece, Sweden, Croatia and Turkey.

This REGALE-project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 956560. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Greece, Germany, France, Spain, Austria, Italy.

Q&A

- Power Management in HPC
- ControlPulp Project: an open-source hardware/software RISC-V controller
- Simulation Framework
- Co-simulation Demo
- **QnA**

