Fine-Grained Application Energy and Power Measurements on the Frontier Exascale System

Oscar Hernandez, **Wael Elwasif**
Oak Ridge National Laboratory

Cray User Group Meeting (CUG 2025)

ORNL IS MANAGED BY UT-BATTELLE LLC
FOR THE US DEPARTMENT OF ENERGY

Motivation

HPC Power Challenges

Increasing power consumption in modern chips (e.g., **AMD MI250X**)

Need for fine-grained power and energy optimizations

Moore’s Law slowdown and efficiency trade-offs

Why Fine-Grained Power Analysis?

Identifying power spikes and inefficiencies

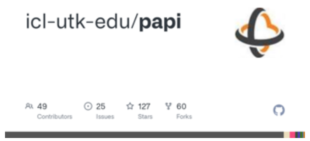
Optimizing application performance and energy use

Key Contributions

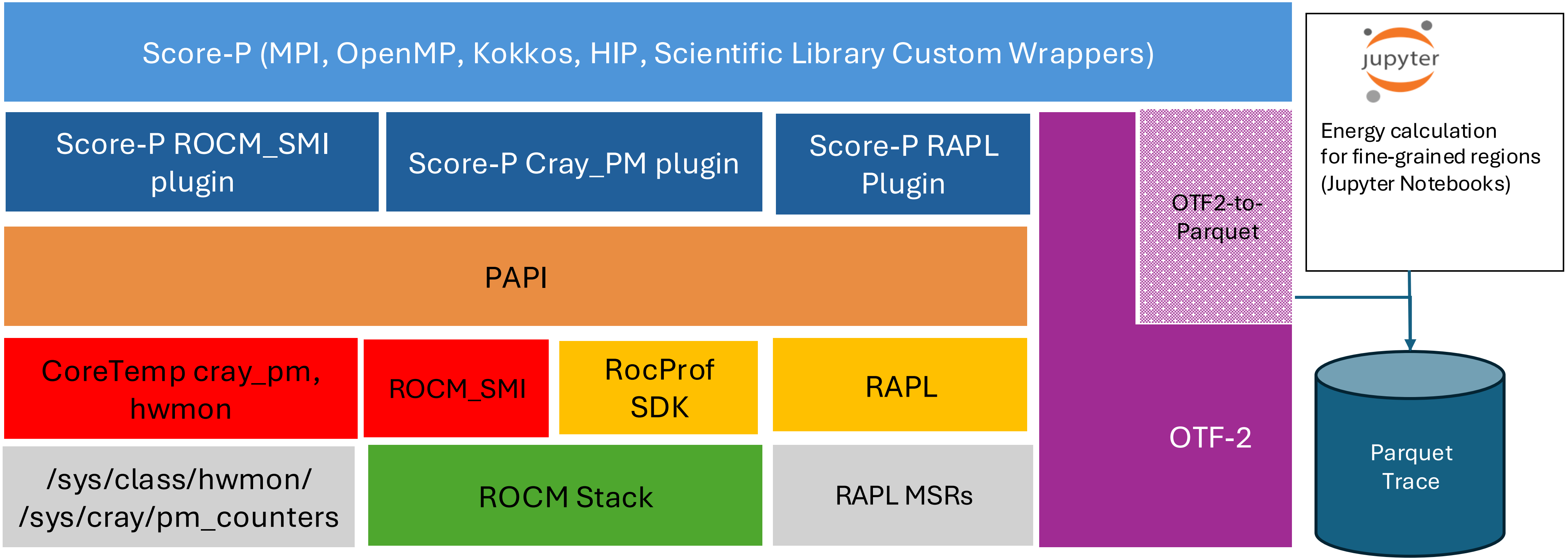
- Development of an open-source toolkit for fine-grained power measurements
- Integration of Linux hwmon, Cray PM, and ROCm SMI with PAPI and Score-P
- High-resolution power sampling (1 ms)
- Mapping power usage to specific application regions

Key Toolkit Components

- Scalable Performance Measurement Infrastructure for Parallel Codes (Score-P)
  https://www.vi-hps.org/projects/score-p/
- Performance Application Programming Interface (PAPI)
  https://icl.utk.edu/papi/
- The Cray Power Management data access for users (PM)
  https://cray-hpe.github.io/docs-csm/en-10/operations/power_management/user_access_to_compute_node_power_data/
- The AMD ROCm system management interface library (ROCm-smi)
  https://github.com/ROCm/rocm_smi_lib
- Vampir performance analysis and visualization
  https://vampir.eu/


icl-utk-edu/**papi**

Performance Analysis Toolkit for Frontier

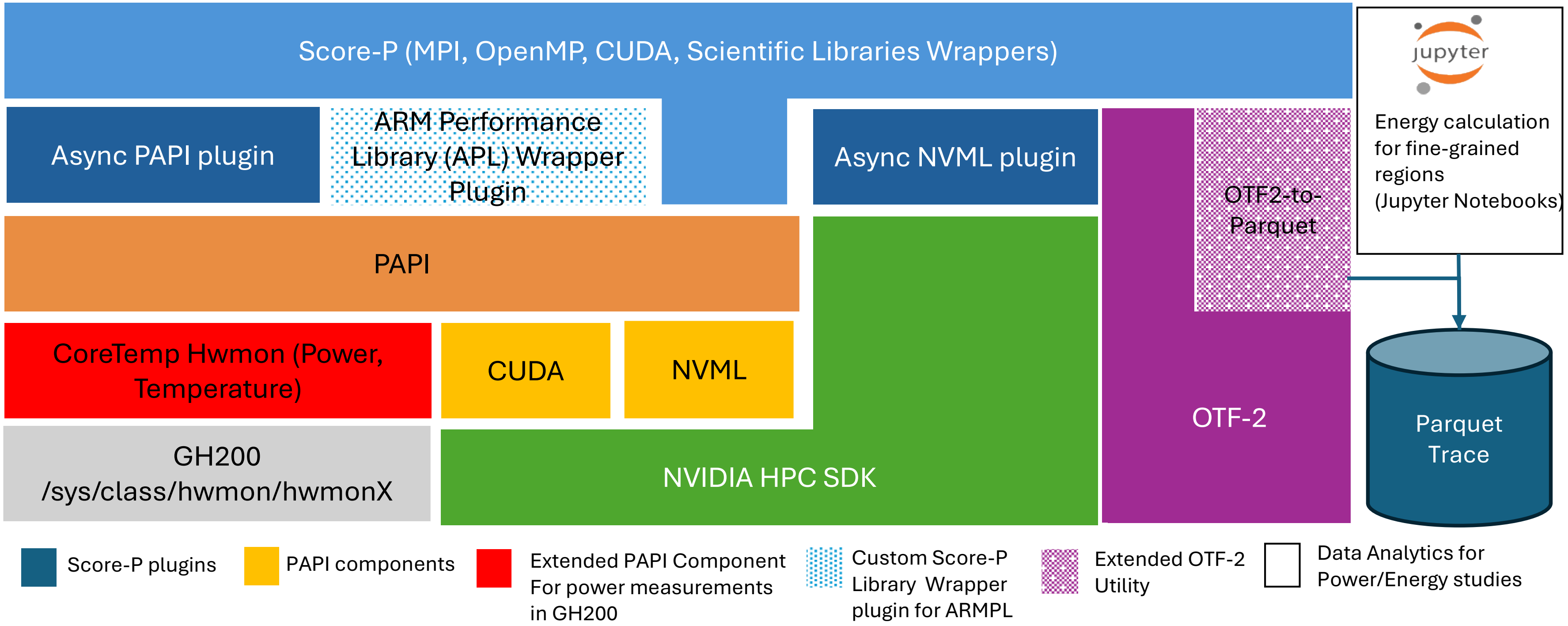


- 🟦 Score-P plugins customized to perform measurements per node.
- 🟨 PAPI components
- 🟥 Extended PAPI Component For power and Energy measurements on Frontier
- ▨ Custom Score-P Library Wrapper plugin for libsci, BLIS/Flame, rocBLAS
- ▨ Extended OTF-2 Utility
- □ Data Analytics for Power/Energy studies

Based on tools from: Hewlett Packard Enterprise, AMD, VAMPIR, Score-P (Scalable performance measurement infrastructure for parallel codes), icl-utk-edu/papi

OAK RIDGE National Laboratory

Performance Analysis Toolkit for GH200



Score-P (MPI, OpenMP, CUDA, Scientific Libraries Wrappers)

Async PAPI plugin

ARM Performance Library (APL) Wrapper Plugin

Async NVML plugin

OTF2-to-Parquet

jupyter

Energy calculation for fine-grained regions (Jupyter Notebooks)

PAPI

CoreTemp Hwmon (Power, Temperature)

CUDA

NVML

OTF-2

Parquet Trace

GH200 /sys/class/hwmon/hwmonX

NVIDIA HPC SDK

Score-P plugins

PAPI components

Extended PAPI Component For power measurements in GH200

Custom Score-P Library Wrapper plugin for ARMPL

Extended OTF-2 Utility

Data Analytics for Power/Energy studies

Python Data Analytics

Extended Components

- PAPI extension for hwmon to measure Superchip and Grace CPU power
- Score-P NVML and PAPI plugins readings
- OTF2-to-Parquet conversion utility
- Jupyter-based analytics for energy profiling

Supported Architectures

- GH200, potential for AMD Instinct and other platforms

Based on tools from:

VAMPIR

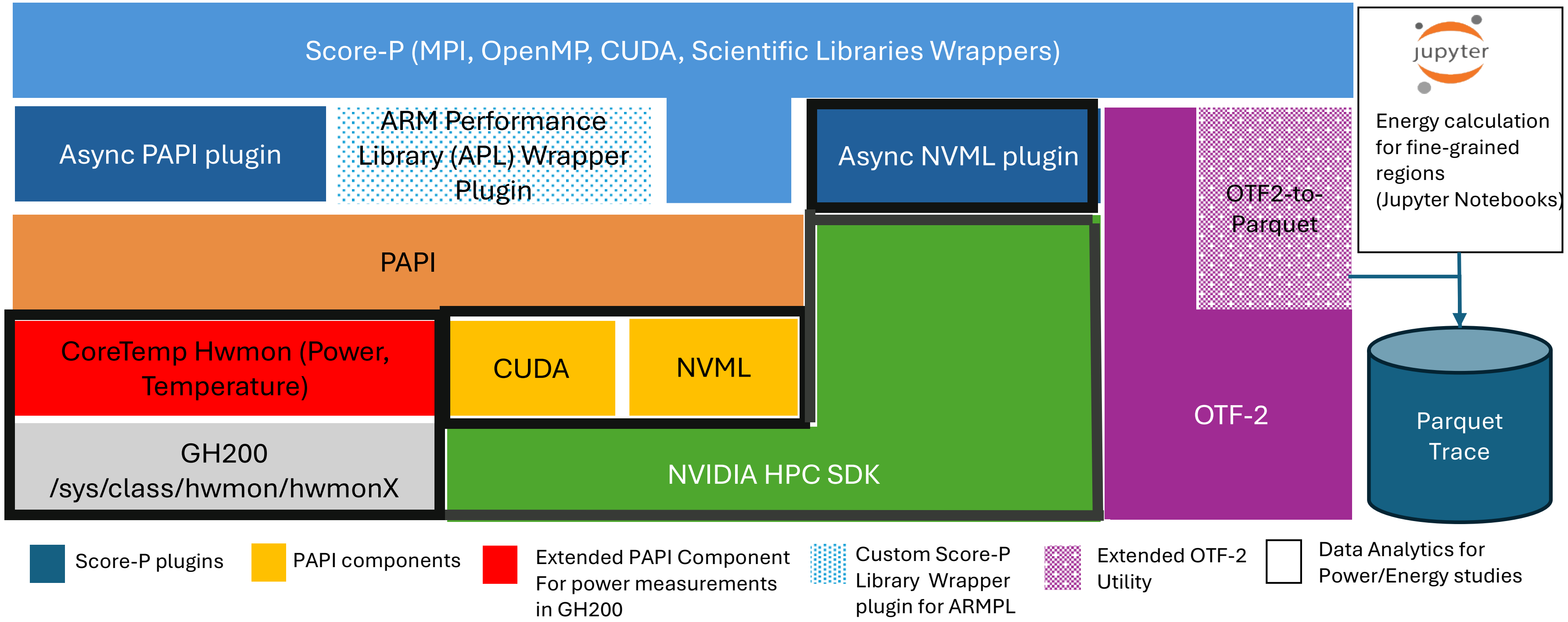
Score-P
Scalable performance measurement infrastructure for parallel codes

icl-utk-edu/papi

NVIDIA

arm

Performance Analysis Toolkit for GH200



Score-P (MPI, OpenMP, CUDA, Scientific Libraries Wrappers)

Async PAPI plugin

ARM Performance Library (APL) Wrapper Plugin

Async NVML plugin

OTF2-to-Parquet

jupyter

Energy calculation for fine-grained regions (Jupyter Notebooks)

PAPI

CoreTemp Hwmon (Power, Temperature)

CUDA

NVML

OTF-2

Parquet Trace

GH200 /sys/class/hwmon/hwmonX

NVIDIA HPC SDK

- Score-P plugins
- PAPI components
- Extended PAPI Component For power measurements in GH200
- Custom Score-P Library Wrapper plugin for ARMPL
- Extended OTF-2 Utility
- □ Data Analytics for Power/Energy studies

HPCAsia 2025, International Workshop on Arm Based HPC. ***Preliminary Study on Fine-Grained Power and Energy Measurements on Grace Hopper GH200 with Open-Source Performance Tools.*** [***https://doi.org/10.1145/3703001.3724383***](https://doi.org/10.1145/3703001.3724383)

Extended Components

- PAPI extension for hwmon to measure Superchip and Grace CPU power
- Score-P NVML and PAPI plugins readings
- OTF2-to-Parquet conversion utility
- Jupyter-based analytics for energy profiling

Supported Architectures

- GH200, potential for AMD Instinct and other platforms

Based on tools from:

VAMPIR

Score-P Scalable performance measurement infrastructure for parallel codes

icl-utk-edu/papi

NVIDIA

arm

PAPI component improvements

Component	Improvement
ROCM_SMI	Added accumulated energy support
Coretemp	Added support for cray_pm counters
RAPL	Support for Milan and Trento CPUs

Score-P plugins customization

Plugin Name	Scope	PAPI component
ROCM_SMI	Per Node	ROCM_SMI
CRAY_PM	Per Node	Coretemp extended to read cray_pm counters
RAPL	Per Node	rapl

Key Benefits

- A unified timeline where power and energy measurements from different libraries, sub-systems, and tools are available.
- Using application trace timeline enables fine-grained analysis of power/energy characteristics of specific code regions
- Open storage format (OTF2) enables third party tools for analysis and insight (though scaling remains a challenge).

User accessible power management (PM) on Cray EX nodes

- Target telemetry and job profiling
- Collection for point-in-time power and energy data
- Made available using sysfs interface `/sys/cray/pm_counters/XYZ`
- Source: HPE User Access to Compute Node Power Data

File	Description
power	Point-in-time power (Watts).
energy	Accumulated energy, in joules.
cpu_power	Point-in-time power (Watts) used by the CPU domain.
cpu_energy	The total energy (Joules) used by the CPU domain.
cpu_temp	Temperature reading (Celsius) of the CPU domain.
memory_power	Point-in-time power (Watts) used by the memory domain.
memory_energy	The total energy (Joules) used by the memory domain.
accel_energy	Accumulated accelerator energy (Joules).
accel_power	Accelerator point-in-time power (Watts).
generation	A counter that increments each time a power cap value is changed.
startup	Startup counter.
freshness	Free-running counter that increments at a rate of approximately 10Hz.
version	Version number for power management counter support.
power_cap	Current power cap limit in Watts; 0 indicates no capping.
raw_scan_hz	The power management scanning rate for all data in pm_counters.

Accessing GPU Power/Energy on Frontier Nodes

Feature	`rsmi_dev_energy_count_get()`	`rsmi_dev_power_ave_get()`	`cray_pm power/energy`
Scope	MI250X GPU	MI250X GPU	MI250X GPU (Blade Controller)
Sampling Interval	1 millisecond	1 millisecond	100 millisecond
Units	Micro Joule	Micro Watt	Watts / Joules
Granularity	High resolution energy counter	Average power over time	Mid-resolution sensors/counters
Accessibility	rocm_smi (amd_smi)	rocm_smi (amd_smi)	`/sys/cray/pm_counters`
Sampling rate	Fixed Interval	Fixed Interval	Fixed Interval
Overhead	API (through sysfs)	API (through sysfs)	User sysfs IO access
Use Case	Fine-grained event correlation	Coarse-grained event correlation	Coarse-grained monitoring / telemetry

Recent rocm_smi library versions include **`rsmi_dev_current_socket_power()`** and **`rsmi_dev_power_get()`** which return average power on the MI250X

Support for Accumulated Energy Monitoring per GPU in ROCm-SMI PAPI component

ROCM_SMI function Integrated:

```
rsmi_status_t rsmi_dev_energy_count_get(uint32_t device, uint64_t *count, float *resolution, uint64_t *timestamp)
```

New Native Event Added:

- Format: `energy_count:device=X`
- Description: *"Accumulated GPU energy, in microjoules (µJ)"*

Integration Points:

- Function pointer registered in dynamic symbol table
- Event handler **access_rsmi_dev_energy_count(...)** implemented to read energy in µJ
- Event appended to event_function_table and registered in htable

Usage:

```
%papi_native_avail | grep energy
| rocm_smi:::energy_count:device=0
|           Accumulated GPU energy, in microjoules (µJ).
```

Counter access:

•Accumulated energy = `(raw_count - offset) × resolution`   `int64_t` in µJ

•Offset refers to the initial value of the energy during initialization, used to start energy accumulation from zero.

Instantaneous GPU Power Estimation using rocm_smi

Methodology

- rocm_smi provides accumulated energy consumption in microjoules (uJ).
- Instantaneous power is approximated using the difference between consecutive energy samples:

$$P_{\text{inst}}^{(i)} \approx \frac{E^{(i)} - E^{(i-1)}}{\Delta t}$$

Sampling Details

- Sampling rate: 1 ms - allows mapping transient power spikes to specific GPU kernels
- Enables correlation of power surges with high-activity regions in application traces

Tradeoffs

- High temporal resolution → More accurate correlation to kernel activity
- May introduce **apparent overestimation** due to sampling noise or short-term transients

Frontier Overview

Architecture

- **9856 Node HPE Cray EX with Slingshot Interconnect**

Key Node Features

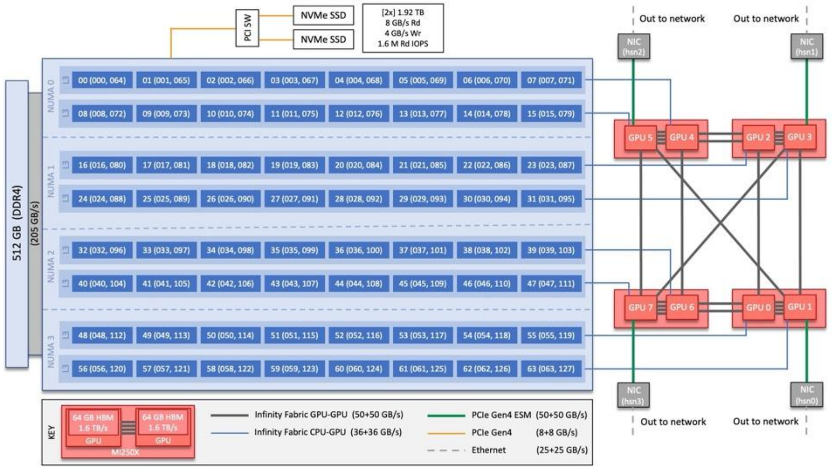
- 64-core AMD EPYC 7A53 CPU (Trento)
- 4x AMD MI250X GPU with 128GB HBM3
    - 2x64 GB GCDS
- Infinity Fabric CPU-GPU and GPU-GPU links

TDP: 500–560W (*GPU*)   225-240W (*CPU*)

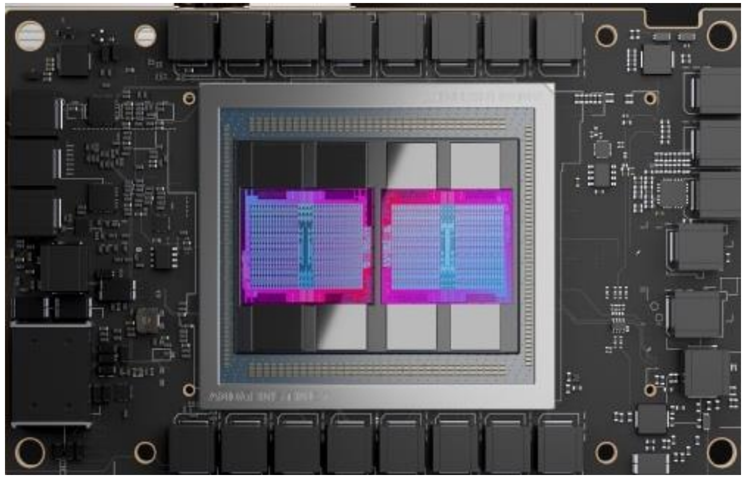
Challenges

- Managing power across CPU, GPU, and I/O
- High-resolution monitoring for transient power spikes

Frontier Node



NUMA 0	L3	00 (000, 064)	01 (001, 065)	02 (002, 066)	03 (003, 067)	04 (004, 068)	05 (005, 069)	06 (006, 070)	07 (007, 071)
	L3	08 (008, 072)	09 (009, 073)	10 (010, 074)	11 (011, 075)	12 (012, 076)	13 (013, 077)	14 (014, 078)	15 (015, 079)
NUMA 1	L3	16 (016, 080)	17 (017, 081)	18 (018, 082)	19 (019, 083)	20 (020, 084)	21 (021, 085)	22 (022, 086)	23 (023, 087)
	L3	24 (024, 088)	25 (025, 089)	26 (026, 090)	27 (027, 091)	28 (028, 092)	29 (029, 093)	30 (030, 094)	31 (031, 095)
NUMA 2	L3	32 (032, 096)	33 (033, 097)	34 (034, 098)	35 (035, 099)	36 (036, 100)	37 (037, 101)	38 (038, 102)	39 (039, 103)
	L3	40 (040, 104)	41 (041, 105)	42 (042, 106)	43 (043, 107)	44 (044, 108)	45 (045, 109)	46 (046, 110)	47 (047, 111)
NUMA 3	L3	48 (048, 112)	49 (049, 113)	50 (050, 114)	51 (051, 115)	52 (052, 116)	53 (053, 117)	54 (054, 118)	55 (055, 119)
	L3	56 (056, 120)	57 (057, 121)	58 (058, 122)	59 (059, 123)	60 (060, 124)	61 (061, 125)	62 (062, 126)	63 (063, 127)



AMD MI250X

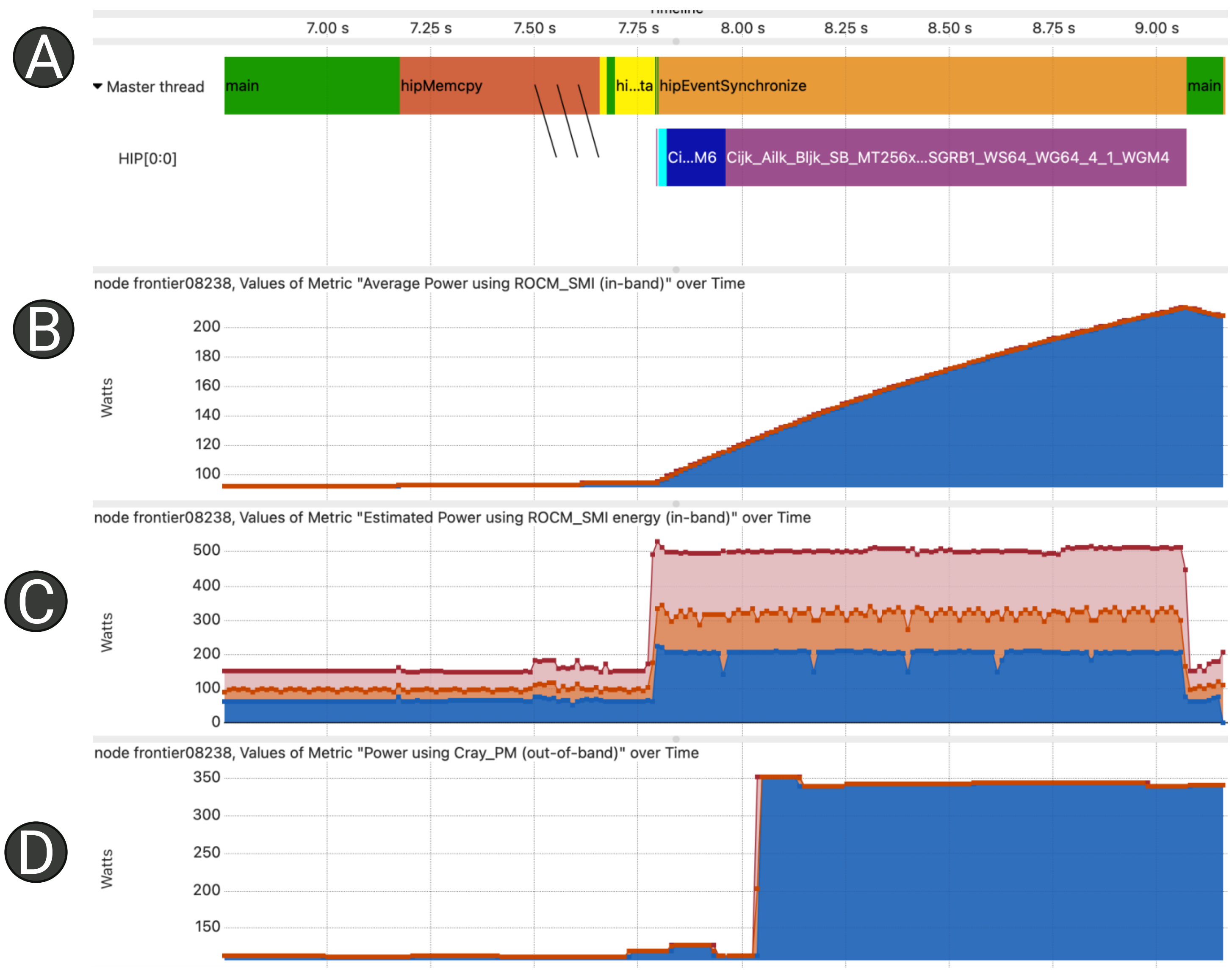
Case Study: GPU SGEMM

- **Application trace timeline**
- CPU initialization phase
- GPU Execution Phase

- **SMI average power measurements**
- Sampling rate : 1 ms
- Ramp up response to instantaneous change in activity

- **SMI instantaneous power estimate**
- Sampling rate 1 ms
- Timely capture of sudden changes
- Min/Max/Average ➔ variability

- **Cray PM**
- Sampling rate : 1 ms
- Delayed response to sudden changes, little/no variability



Case Study : CPU DGEMM with OpenMP

Sampling Rate : 1 ms

CPU thread activity timeline
Barriers do not consume much energy

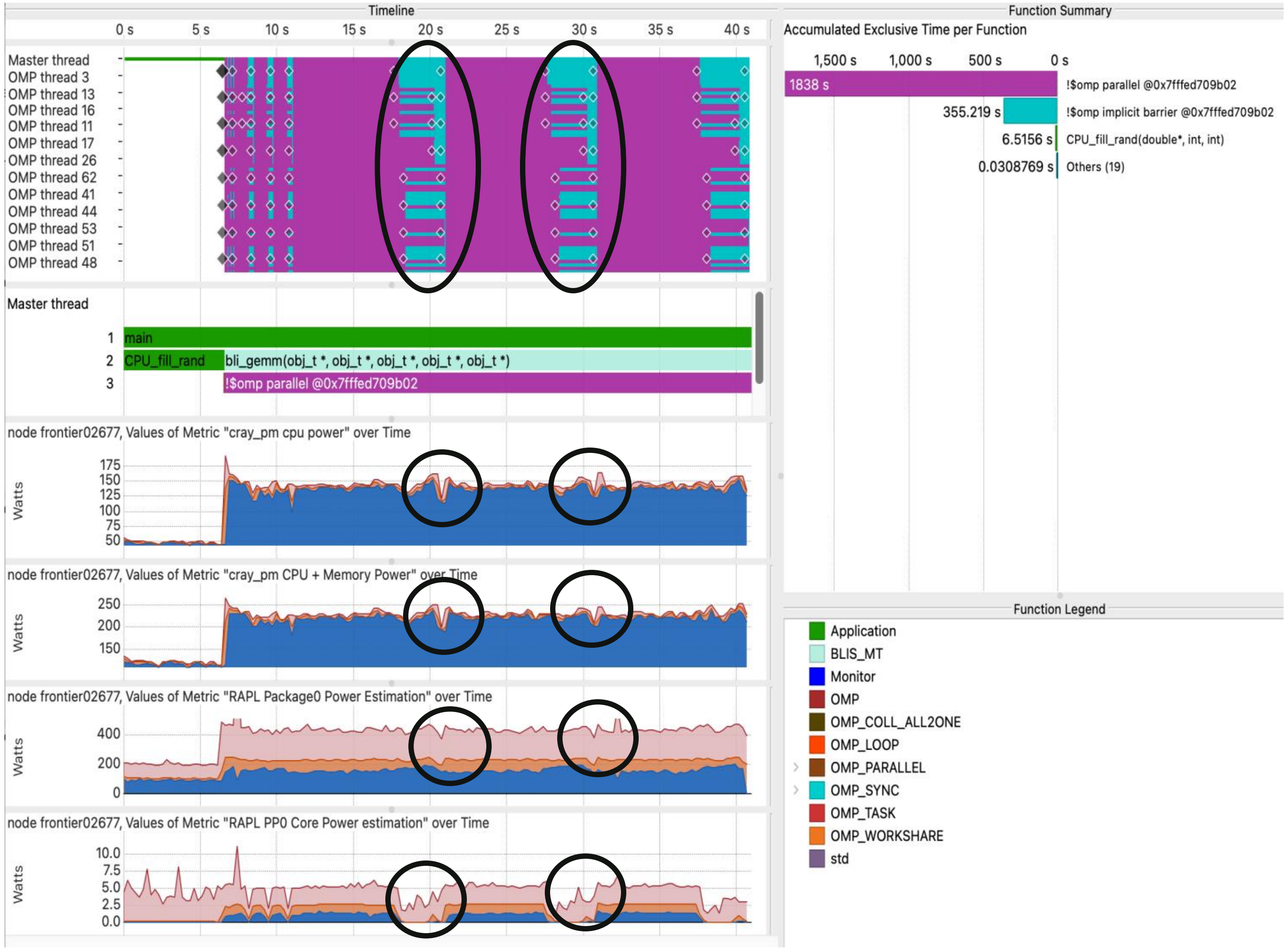
Process Timeline

Cray PM CPU

Cray PM CPU + Memory

RAPL Package0

RAPL PP0


Timeline
0 s
5 s
10 s
15 s
20 s
25 s
30 s
35 s
40 s
Master thread
OMP thread 3
OMP thread 13
OMP thread 16
OMP thread 11
OMP thread 17
OMP thread 26
OMP thread 62
OMP thread 41
OMP thread 44
OMP thread 53
OMP thread 51
OMP thread 48
Master thread
1 main
2 CPU_fill_rand
bli_gemm(obj_t *, obj_t *, obj_t *, obj_t *, obj_t *)
3 !$omp parallel @0x7fffed709b02
node frontier02677, Values of Metric "cray_pm cpu power" over Time
Watts
175
150
125
100
75
50
node frontier02677, Values of Metric "cray_pm CPU + Memory Power" over Time
Watts
250
200
150
node frontier02677, Values of Metric "RAPL Package0 Power Estimation" over Time
Watts
400
200
0
node frontier02677, Values of Metric "RAPL PP0 Core Power estimation" over Time
Watts
10.0
7.5
5.0
2.5
0.0
Function Summary
Accumulated Exclusive Time per Function
1,500 s
1,000 s
500 s
0 s
1838 s
!$omp parallel @0x7fffed709b02
355.219 s
!$omp implicit barrier @0x7fffed709b02
6.5156 s
CPU_fill_rand(double*, int, int)
0.0308769 s
Others (19)
Function Legend
Application
BLIS_MT
Monitor
OMP
OMP_COLL_ALL2ONE
OMP_LOOP
OMP_PARALLEL
OMP_SYNC
OMP_TASK
OMP_WORKSHARE
std

Case Study: rocHPL 6.0.0 (Single rank, 7 OpenMP threads)

Sampling Rate : 1 ms

Process timeline

Cray PM node power

RAPL Package Power

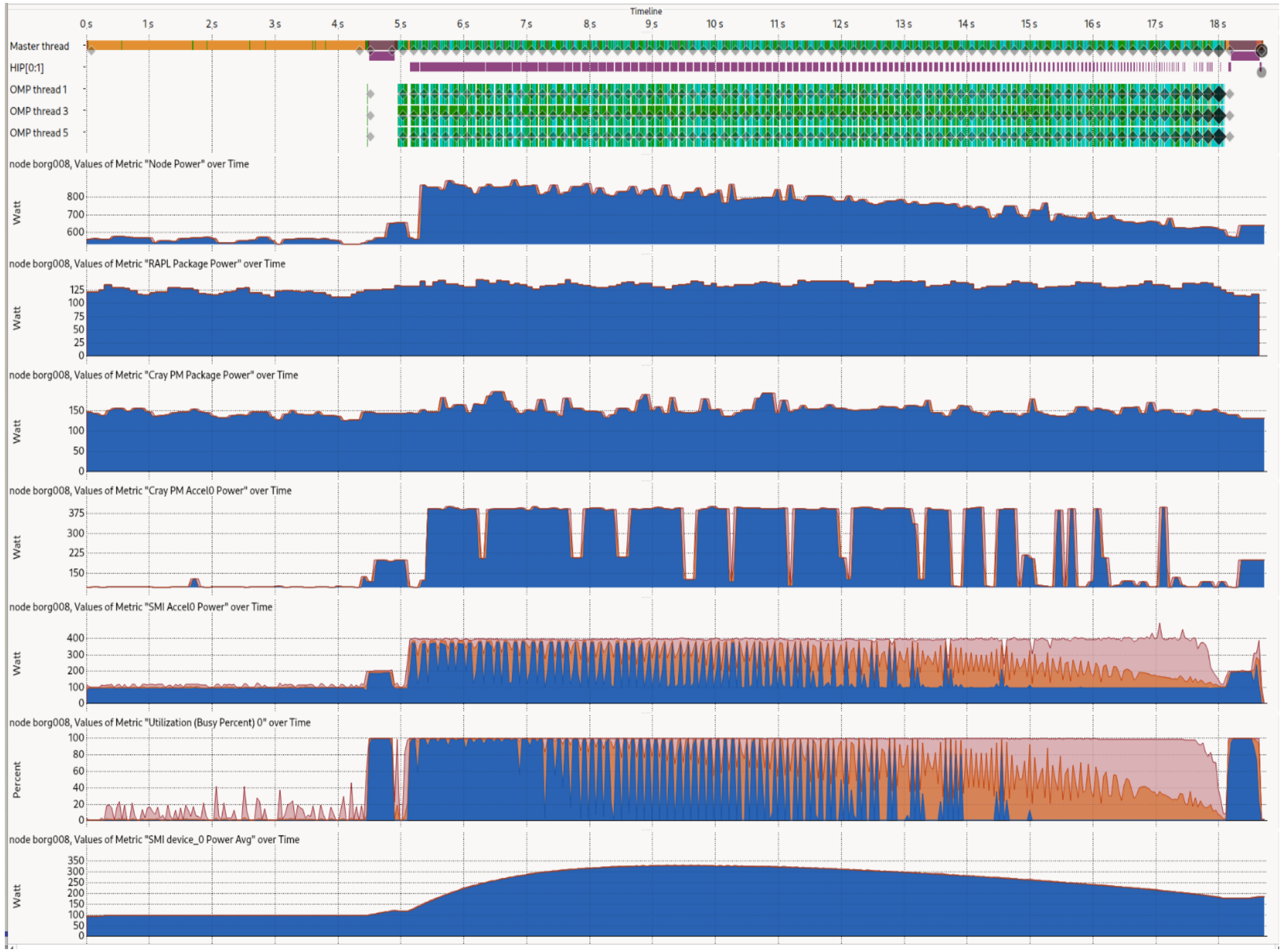
Cray PM Package (CPU + Memory)

RAPL Package0

SMI Instantaneous power estimate

SMI GPU utilization

SMI average power


Timeline
0 s
1 s
2 s
3 s
4 s
5 s
6 s
7 s
8 s
9 s
10 s
11 s
12 s
13 s
14 s
15 s
16 s
17 s
18 s
Master thread
HIP[0:1]
OMP thread 1
OMP thread 3
OMP thread 5
node borg008, Values of Metric "Node Power" over Time
Watt
800
700
600
node borg008, Values of Metric "RAPL Package Power" over Time
Watt
125
100
75
50
25
0
node borg008, Values of Metric "Cray PM Package Power" over Time
Watt
150
100
50
0
node borg008, Values of Metric "Cray PM Accel0 Power" over Time
Watt
375
300
225
150
node borg008, Values of Metric "SMI Accel0 Power" over Time
Watt
400
300
200
100
0
node borg008, Values of Metric "Utilization (Busy Percent) 0" over Time
Percent
100
80
60
40
20
0
node borg008, Values of Metric "SMI device_0 Power Avg" over Time
Watt
350
300
250
200
150
100
50
0

Case Study : 8 rank rocHPL x 7 OpenMP threads

Process timeline

Cray PM node power

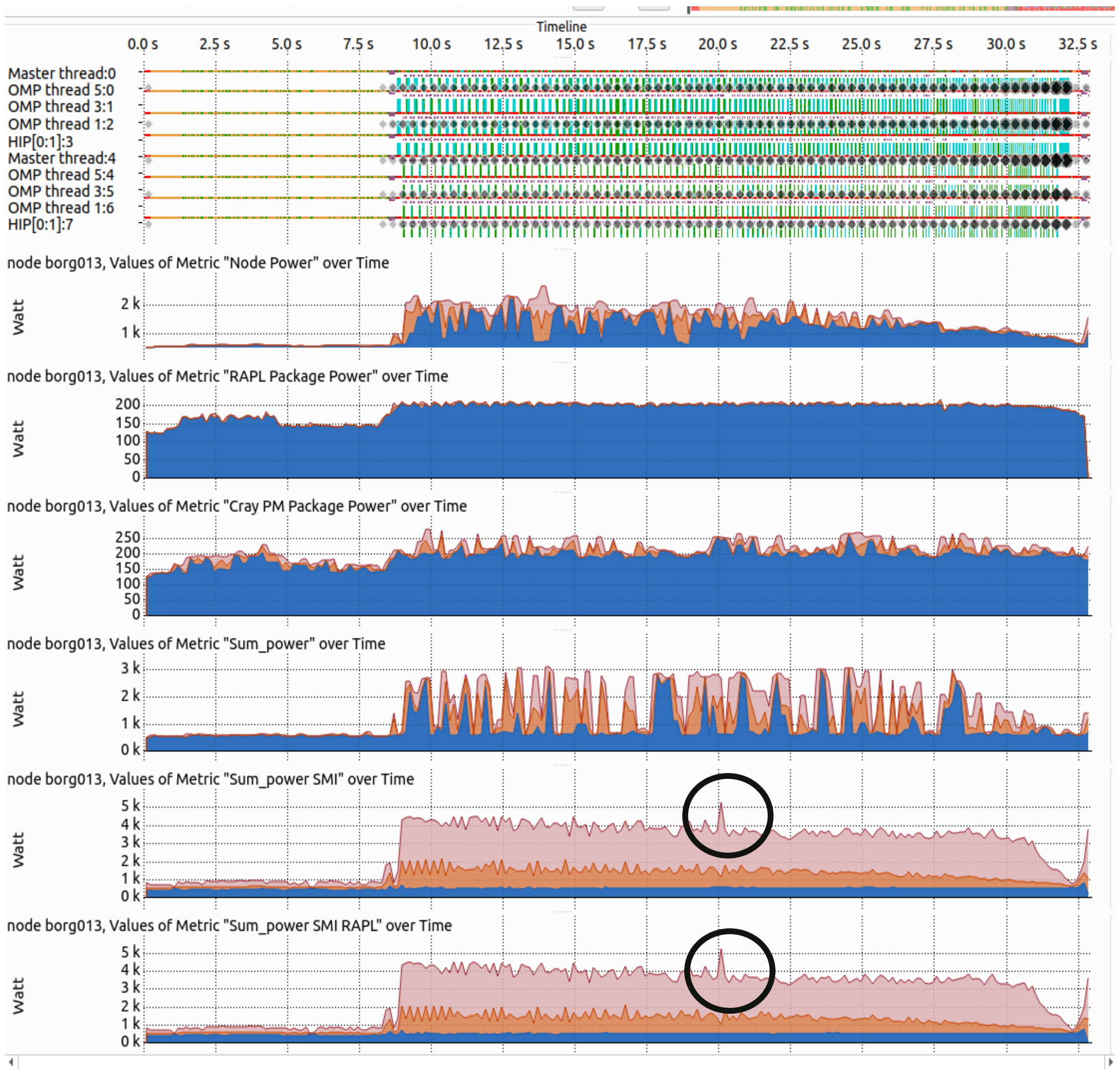
RAPL Package Power

Cray PM Package (CPU + Memory)

Cray PM Sum Component Power

Sum SMI power + Cray PM memory + Cray PM CPU

Sum SMI power + RAPL Package Power


Timeline
0.0 s
2.5 s
5.0 s
7.5 s
10.0 s
12.5 s
15.0 s
17.5 s
20.0 s
22.5 s
25.0 s
27.5 s
30.0 s
32.5 s
Master thread:0
OMP thread 5:0
OMP thread 3:1
OMP thread 1:2
HIP[0:1]:3
Master thread:4
OMP thread 5:4
OMP thread 3:5
OMP thread 1:6
HIP[0:1]:7
node borg013, Values of Metric "Node Power" over Time
Watt
2 k
1 k
node borg013, Values of Metric "RAPL Package Power" over Time
Watt
200
150
100
50
0
node borg013, Values of Metric "Cray PM Package Power" over Time
Watt
250
200
150
100
50
0
node borg013, Values of Metric "Sum_power" over Time
Watt
3 k
2 k
1 k
0 k
node borg013, Values of Metric "Sum_power SMI" over Time
Watt
5 k
4 k
3 k
2 k
1 k
0 k
node borg013, Values of Metric "Sum_power SMI RAPL" over Time
Watt
5 k
4 k
3 k
2 k
1 k
0 k

Case Study : 8 rank rocHPL x 7 OpenMP threads

Process timeline

Cray PM node power

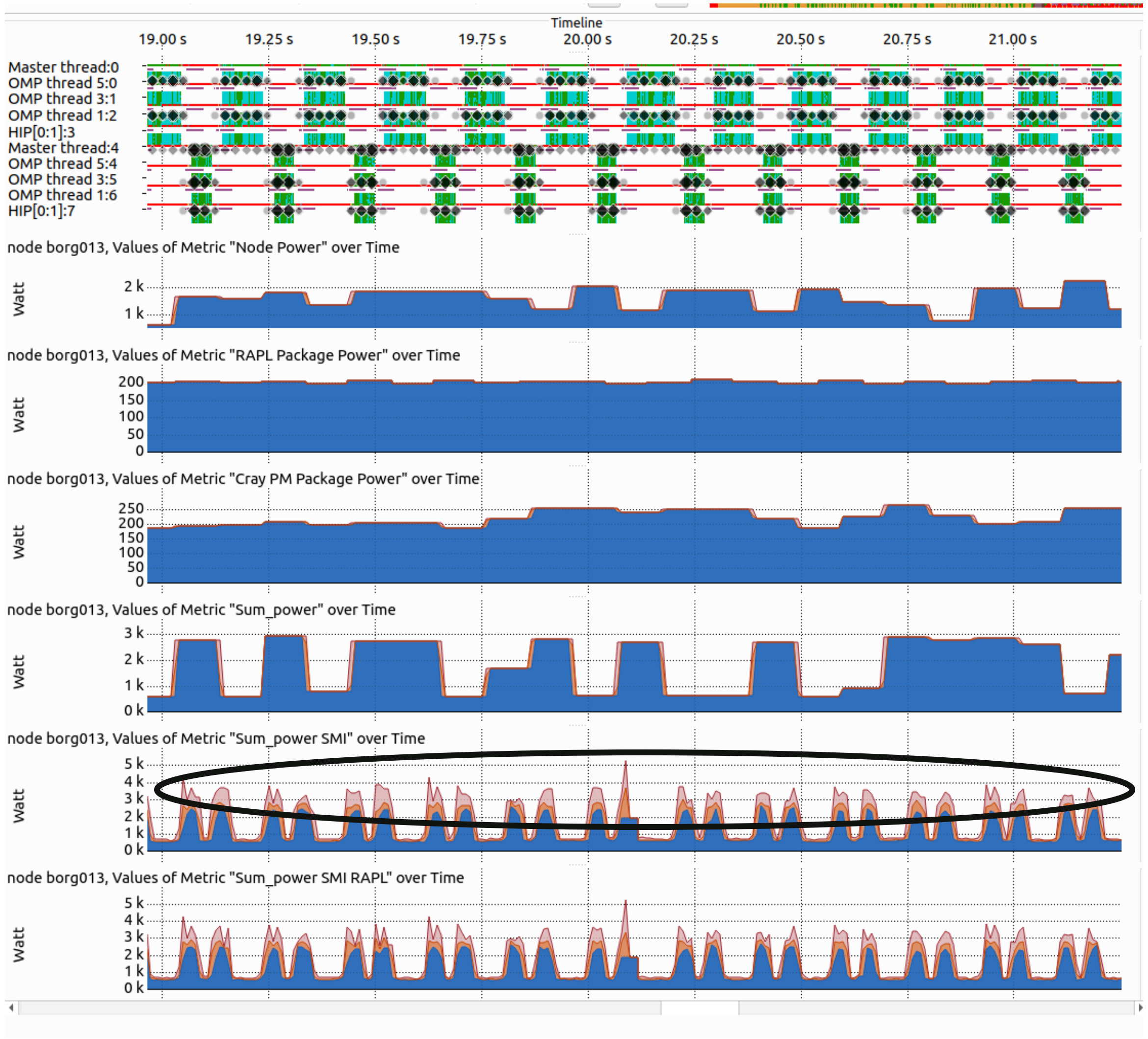
RAPL Package Power

Cray PM Package (CPU + Memory)

Cray PM Sum Component Power

Sum SMI power + Cray PM memory + Cray PM CPU

Sum SMI power + RAPL Package Power


Timeline
19.00 s
19.25 s
19.50 s
19.75 s
20.00 s
20.25 s
20.50 s
20.75 s
21.00 s
Master thread:0
OMP thread 5:0
OMP thread 3:1
OMP thread 1:2
HIP[0:1]:3
Master thread:4
OMP thread 5:4
OMP thread 3:5
OMP thread 1:6
HIP[0:1]:7
node borg013, Values of Metric "Node Power" over Time
Watt
2 k
1 k
node borg013, Values of Metric "RAPL Package Power" over Time
Watt
200
150
100
50
0
node borg013, Values of Metric "Cray PM Package Power" over Time
Watt
250
200
150
100
50
0
node borg013, Values of Metric "Sum_power" over Time
Watt
3 k
2 k
1 k
0 k
node borg013, Values of Metric "Sum_power SMI" over Time
Watt
5 k
4 k
3 k
2 k
1 k
0 k
node borg013, Values of Metric "Sum_power SMI RAPL" over Time
Watt
5 k
4 k
3 k
2 k
1 k
0 k

Conclusions & Further work

- Toolkit for fine grained application energy and power analysis
  - Combines Score-P, Cray PM, ROCm SMI, and PAPI
  - Single timeline for in-band and out-of-band measurements
  - Capture rapid changes in power usage, attribute to code regions
  - Foundation for detailed energy analysis and optimizations
- Further investigations and questions
  - Scalable analytics for large scale OTF2 traces
  - RAPL readings for core energy consumption
  - Investigate energy dependence on performance counters
  - Cray PM aggregate node power vs sum of individual components

Acknowledgement

This research used resources from the Oak Ridge Leadership Computing Facility, which is a US Department of Energy (DOE) Office of Science user facility supported under contract DE-AC05-00OR22725. This work was also supported by the DOE Office of Science, Advanced Scientific Computing Research, Express project “Leveraging Open- Source Simulators to Enable hw/sw Co-design of Next-generation HPC Systems” (DE-FOA-0002950).

2025 OLCF User Meeting

📅 August 5-6, 2025

- Facility updates & expert talks
- Deep-dive discussions with OLCF staff
- Networking with top computational scientists

Register Today!

www.olcf.ornl.gov/olcf-user-meeting

OAK RIDGE National Laboratory

A data workflow to refine bio-macromolecular crystals using machin
Leveraging GNU Parallel for Optimal Utilization of HPC Resources
Molecular Modeling of

Questions?