Centre Informatique National de l'Enseignement Supérieur

HPC

IT Hosting

Archiving

Running Energy-Efficient HPL on APUs: Strategies and Best Practices

EESP Workshop 2025

June 13, 2025

Gabriel Hautreux (CINES)
Head of HPC Dep.
hautreux@cines.fr

Jean-Yves Vet (HPE)
Application Performance Engineer
vet@hpe.com

Adastra-2

Extension Installed in 2024

- **1 HPE Cray EX400 cabinet**
  - **14 HPE Cray EX255a blades**
    - **28 accelerated compute nodes (2 per blade)**
      - **Each node**

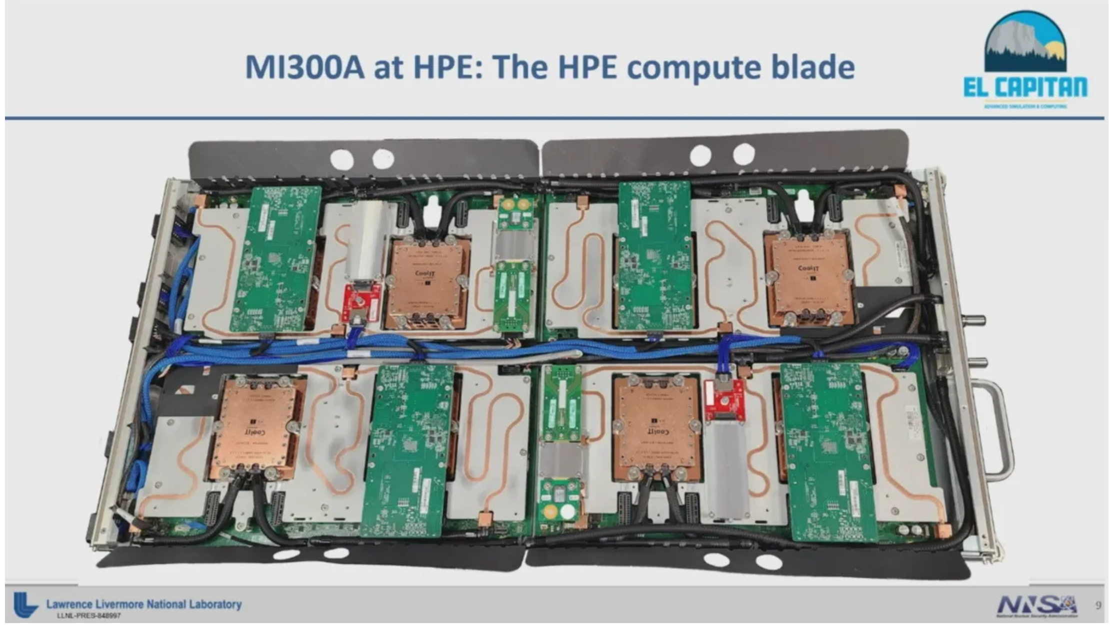
        - **4** AMD Instinct MI300A (APUs)

        - **4** 200Gbps Slingshot NICs

Same tech as ***El Capitan.***

Fanless direct liquid cooling

- Dissipates 97% of the heat
- Input water at 30°C (86 °F)



MI300A at HPE: The HPE compute blade

EL CAPITAN

Lawrence Livermore National Laboratory

LLNL-PRES-848997

NNSA

9

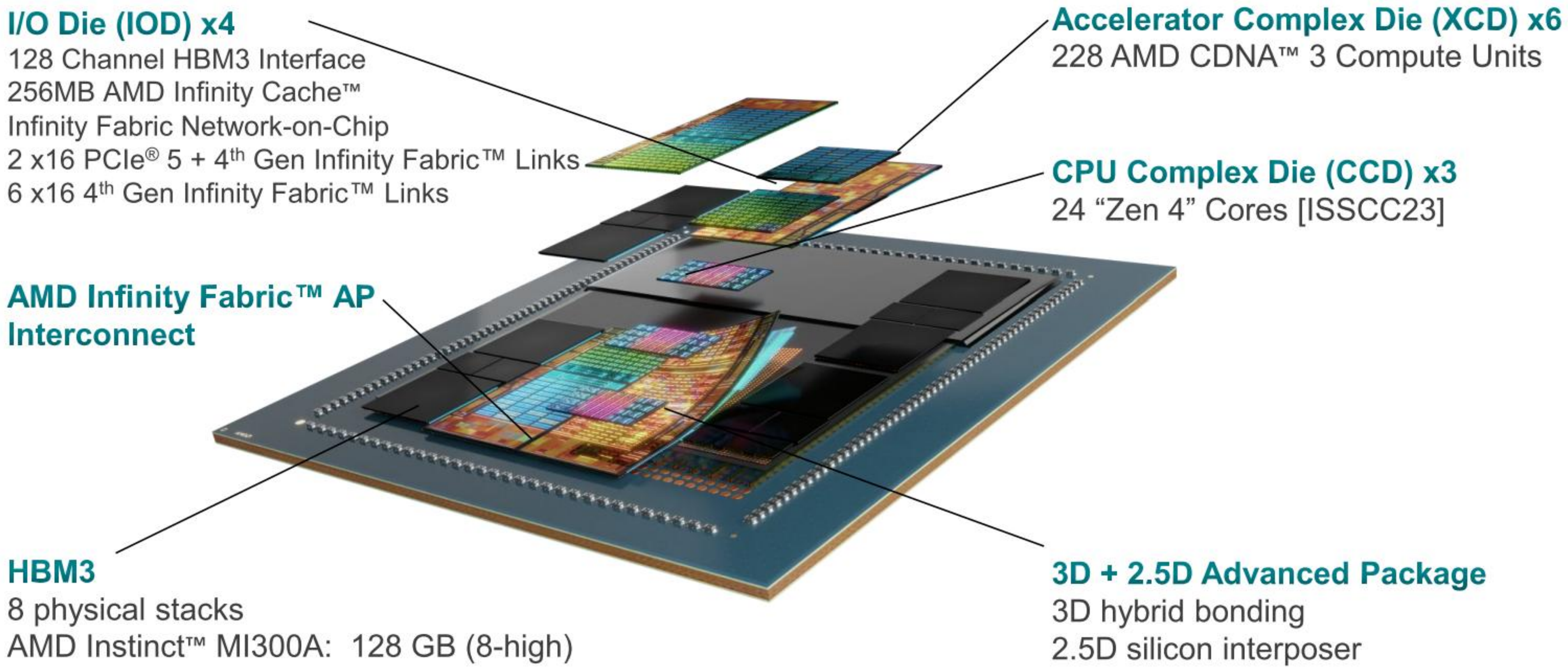
Image credit: LLNL, ISC 2023

Rankings

	Nov 2024	June 2025
The GREEN 500	**3**	**3**

AMD Instinct MI300A

Modular Chiplet Package



I/O Die (IOD) x4
128 Channel HBM3 Interface
256MB AMD Infinity Cache™
Infinity Fabric Network-on-Chip
2 x16 PCIe® 5 + 4th Gen Infinity Fabric™ Links
6 x16 4th Gen Infinity Fabric™ Links

Accelerator Complex Die (XCD) x6
228 AMD CDNA™ 3 Compute Units

CPU Complex Die (CCD) x3
24 “Zen 4” Cores [ISSCC23]

AMD Infinity Fabric™ AP Interconnect

HBM3
8 physical stacks
AMD Instinct™ MI300A: 128 GB (8-high)

3D + 2.5D Advanced Package
3D hybrid bonding
2.5D silicon interposer

Image credit: AMD

Aim of the Paper and Agenda Overview

Objective

Explore universal methods and APU-based strategies to enhance energy efficiency in HPL

Universal Optimization Strategies

- Optimize the use of available memory
- Assess and minimize system noise
- Perform node selection

Strategies for APU-Based Systems

- Initiate memory compaction to reduce fragmentation
- Power Management Strategies for APUs

Universal Optimization Strategies

1- Optimize the Use of Available Memory

Getting more usable memory for larger matrix with HPL

Unbalanced free memory on NUMA nodes due to standard Linux practices

	NUMA 0	NUMA 1	NUMA 2	NUMA 3
Size (MB)	**127811**	128719	128678	128706
Free (MB)	**125468**	127104	127221	126464

Over 1GB is unavailable on NUMA node 0 compared to other NUMA nodes, resulting in more than 4GB being unusable with HPL

Strategies

- Temporarily disable *kdump* reserving ~400 MB on NUMA 0.
- Remount in-memory filesystems with page interleaving skipping NUMA 0

Outcome

Recovering ~4GB for the application

Universal Optimization Strategies

2- Assess and Minimize System Noise

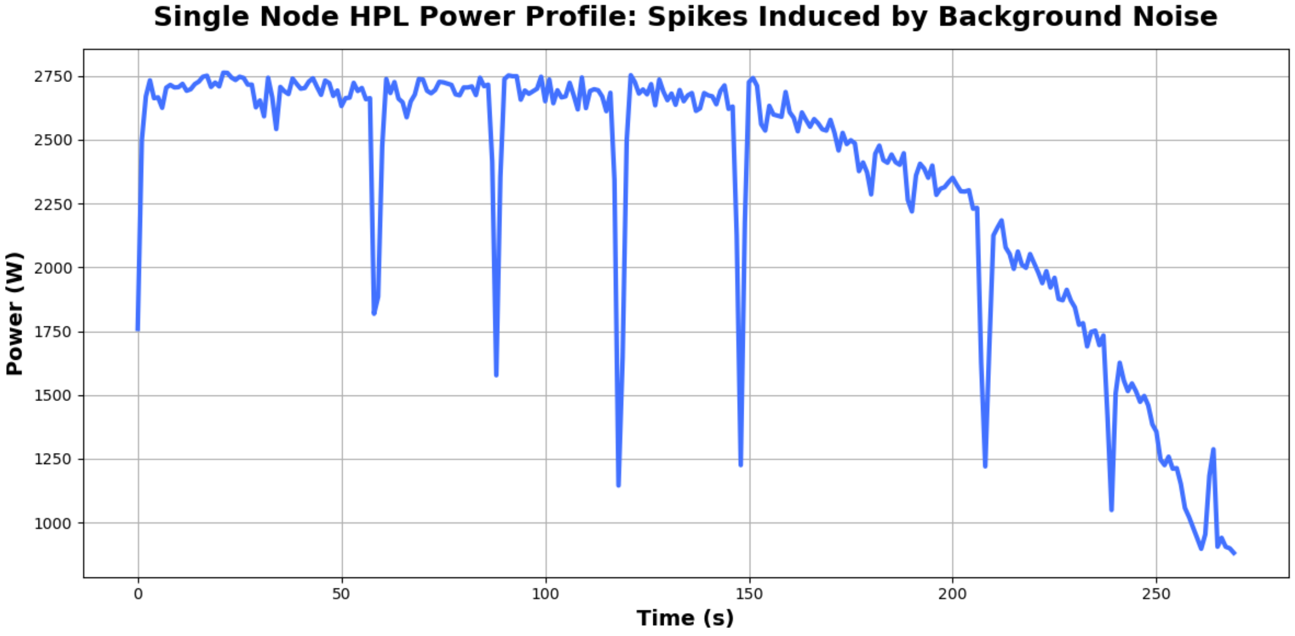
Tracking the progression of the energy counter every second

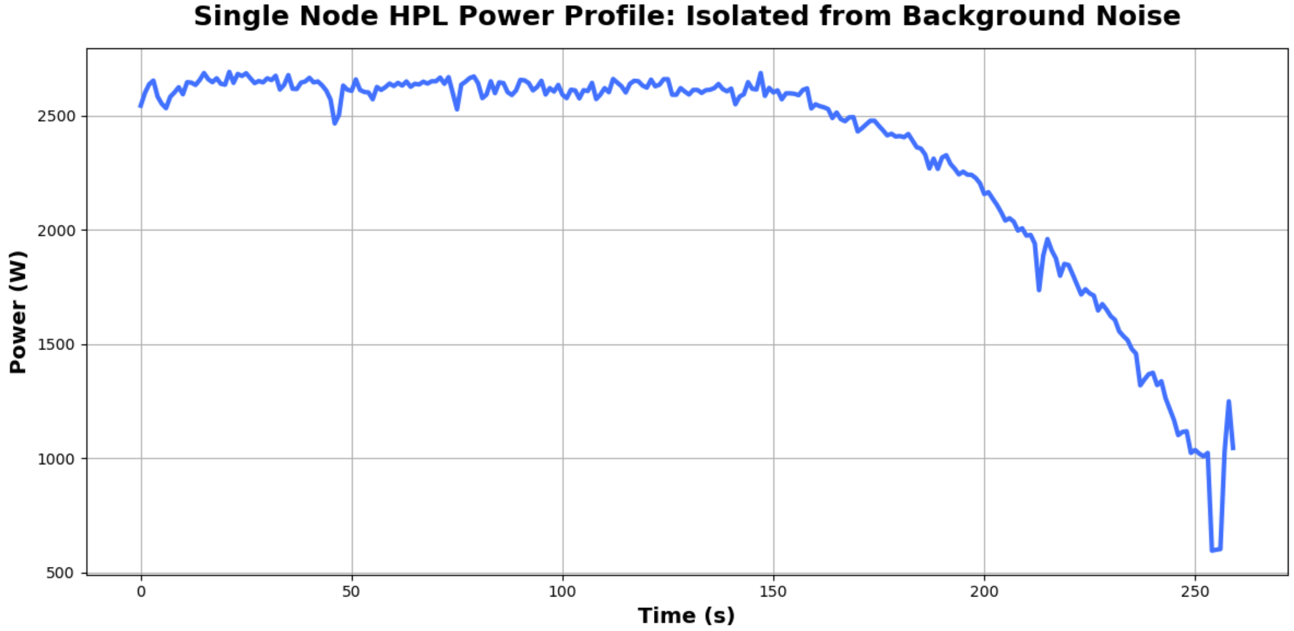
Drop in efficiency every 30s with more than one node

- Caused by Slurm *acct_gather* plugin
- Need more detailed analysis to pinpoint the root cause

Other strategies

- Disabling the *Address Space Layout Randomization* (ASLR) for more perf. stability
- IRQs bound to isolated hardware threads associated with cores not used by HPL





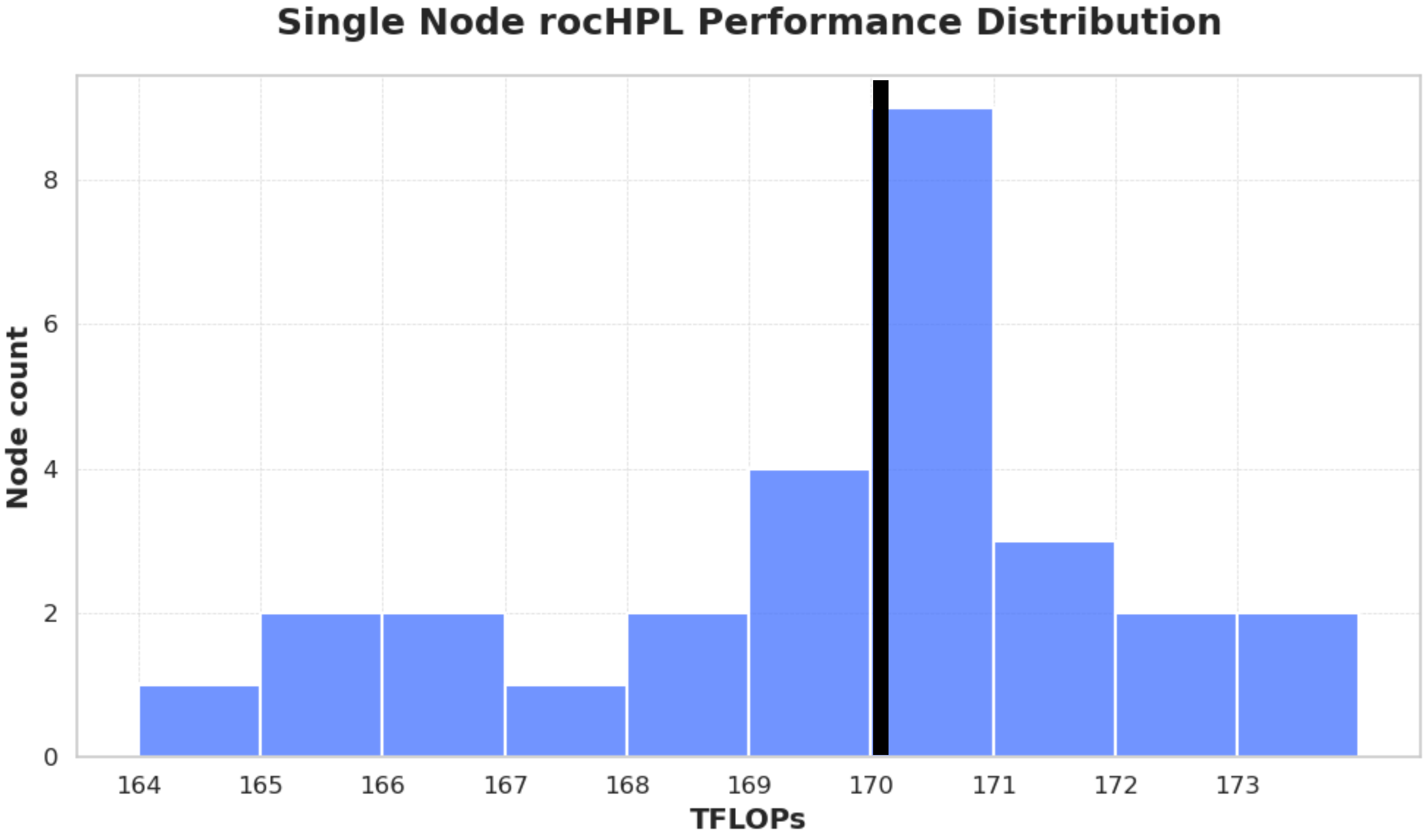
Universal Optimization Strategies

3- Perform Node Selection

5% variation in performance and efficiency between nodes

Strategies

- Select nodes that achieve over 170 TFLOPs for performance runs
- Choose nodes that maximize GFLOPs/W for efficiency runs



Strategies for APU-Based Systems

Memory Implications in APU Systems

5% Unified memory between CPU and GPU accelerates fragmentation

Up to a 5x slowdown with large matrices

Strategies

- Initiate memory compaction at the end of each job and wait a few minutes

Matrix size (N)	Compaction (before the run)	HPL result (TFLOPs)
230400	no	167.6
243072	no	34.8
243072	yes	173.1

Single node HPL performance before and after memory compaction

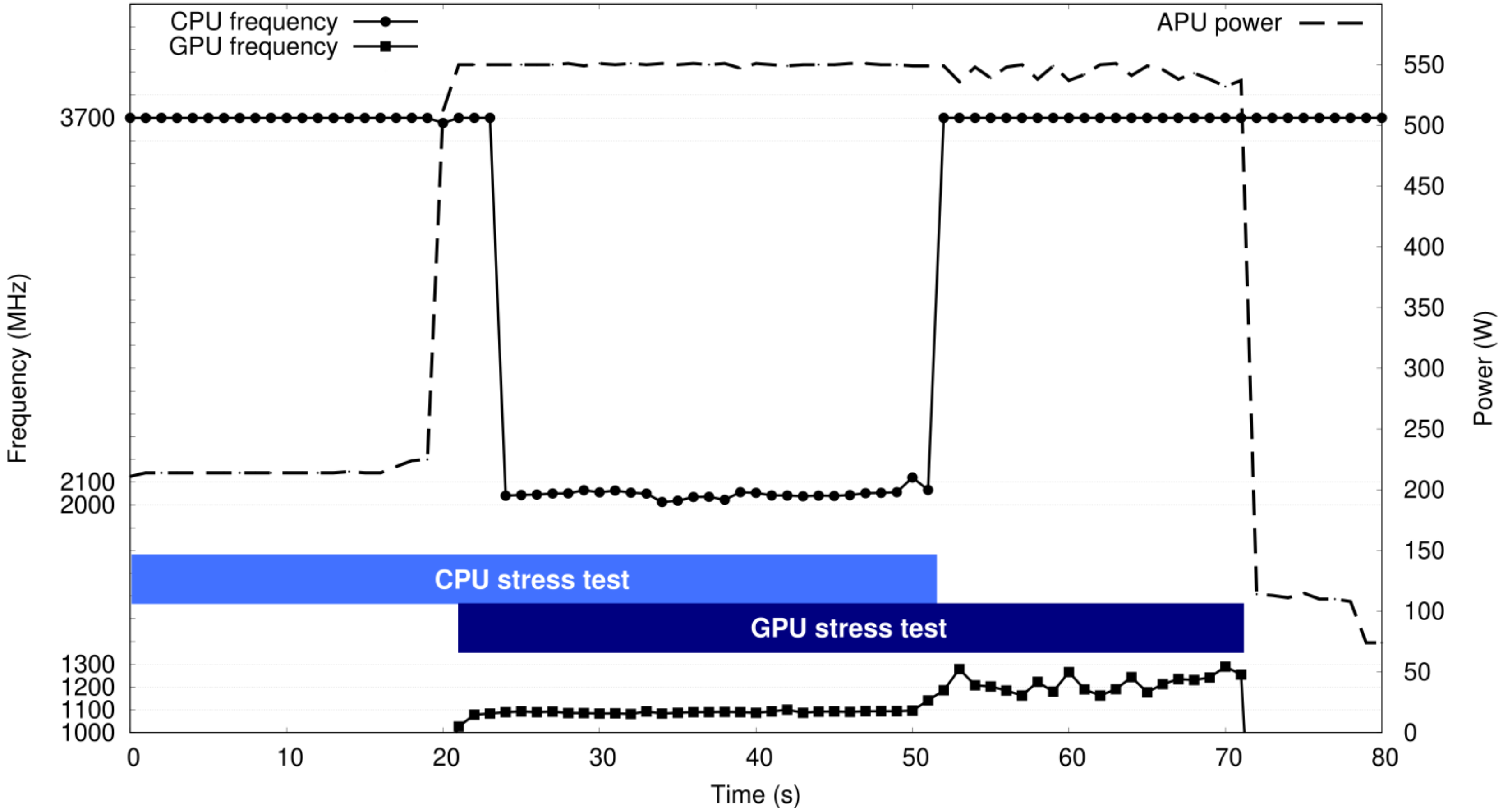
Strategies for APU-Based Systems

CPU, GPU and HBM Sharing the Same Power Envelop

CPU-GPU interdependency

- Package power limit: **550W**
- GPU performance takes precedence over CPU
- Significant variation in CPU frequency

We need to identify methods to optimize CPU-GPU priority arbitration

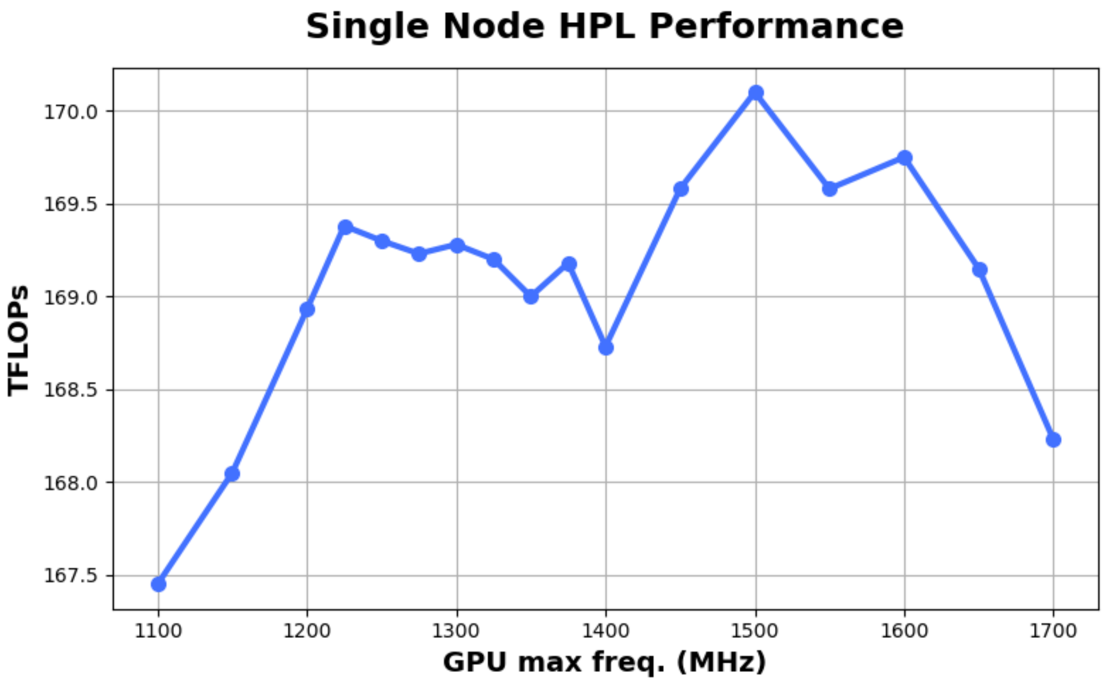


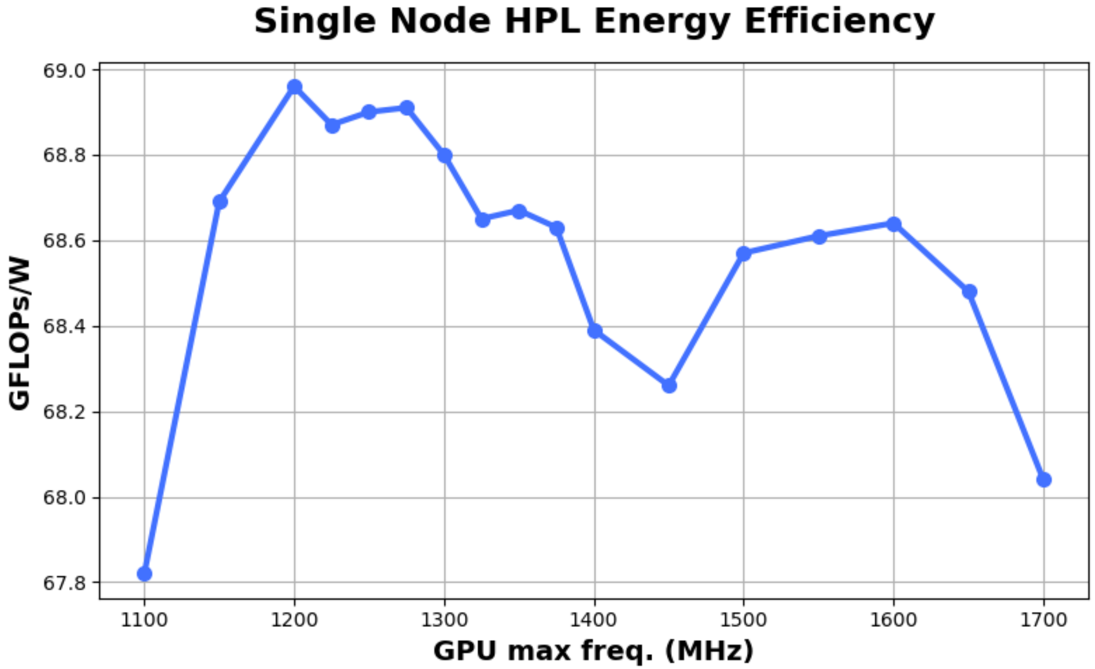
Strategies for APU-Based Systems

1- GPU Frequency Capping

An effective way of distributing power usage between CPU and GPU

- Highest computational throughput was obtained at 1500 MHz
- Optimal frequency for energy efficiency was between 1200 MHz and 1300 MHz
- Selected **1275 MHz** as achieving slightly better results in 16-node runs



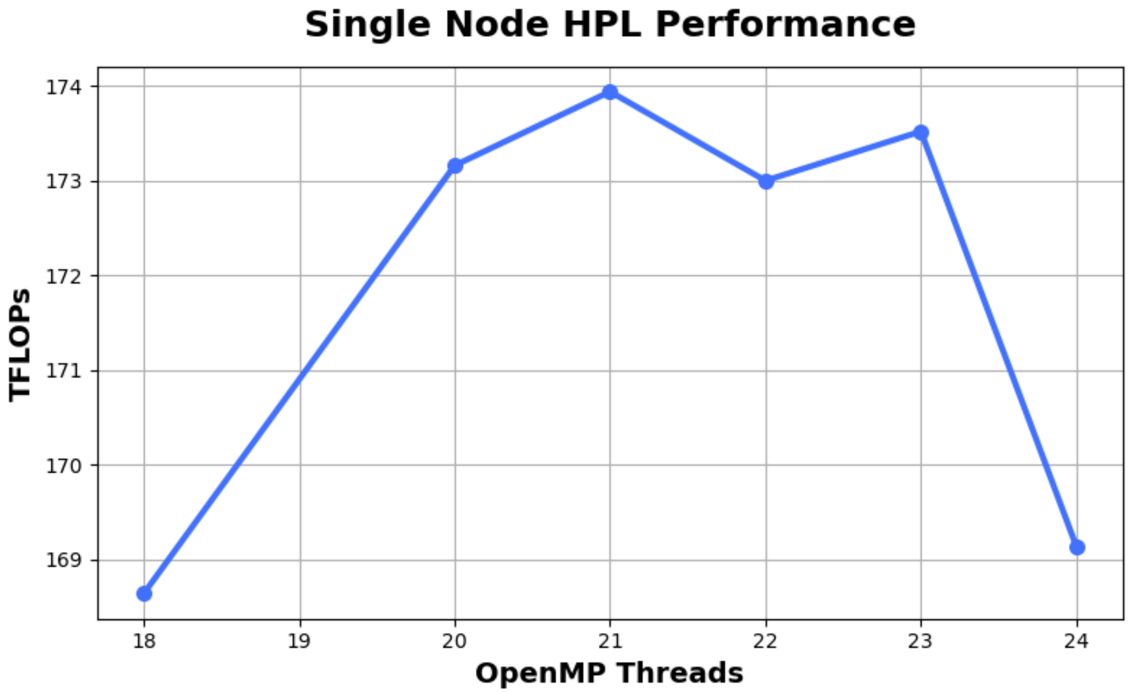


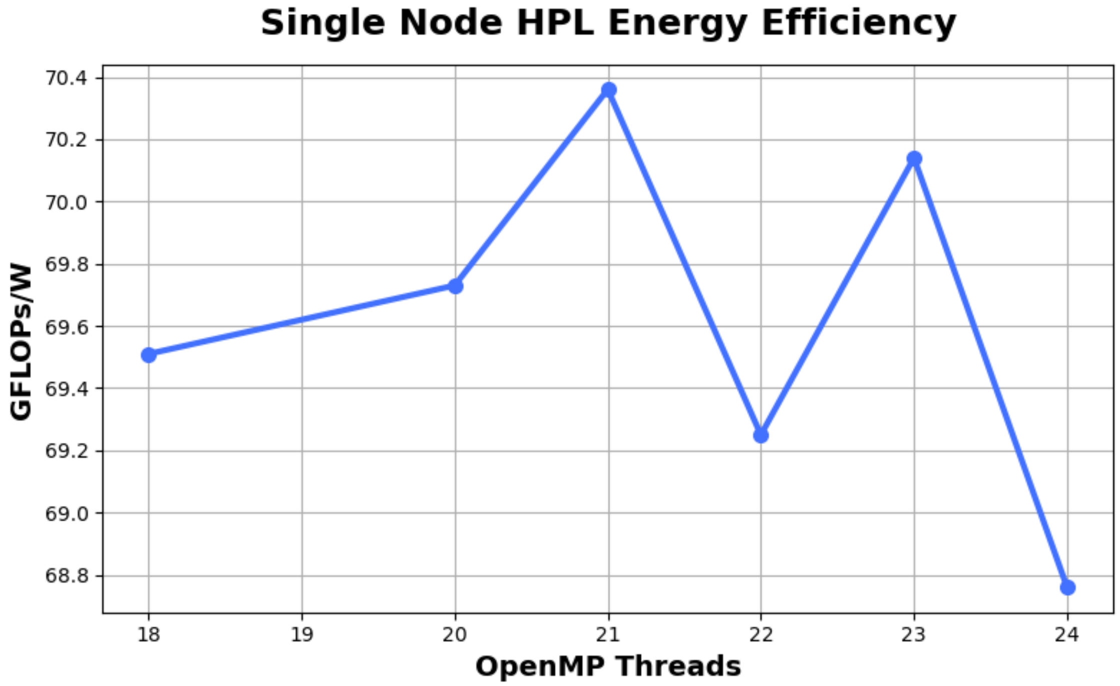
Strategies for APU-Based Systems

2- Quantity of CPU Cores

Using all 24 CPU cores per APU leads to performance degradation

- Might be caused by resource contention between HPL and *amdgpu* IRQs
- **23 cores** seems to minimize interferences
- But using **21 cores is best for overall performance and energy efficiency:** Probably due to reduced CPU power consumption and allowing more power for the GPU part





Strategies for APU-Based Systems

3- APU Power Capping

Capping APU power to improves energy efficiency in some cases

- **Optimal combination:** 530W power cap with 1275MHz GPU freq. cap for HPL.
- **Efficiency gain:** up to 1 GFLOPs/W

Conclusion

Universal Strategies

- Enhance performance stability
- Be able to utilize larger matrices with HPL

APU-Specific Strategies

- Address fragmentation (compaction needed)
- Power budget shared between CPU, GPU and HBM
  - Reducing GPU frequency significantly enhances energy efficiency and performance with the MI300A and HPL workload
  - Keeping a few CPU cores unused can further improve GPU performance and energy efficiency

Future work

- Examine the differences in power efficiency among various discrete APUs of the same type, noting that a variation of up to 15% was observed with the MI300A
- Evaluate the effectiveness of simple models (shifting priority towards CPU at the end of the HPL run) vs. dynamic approaches for frequency and power capping

Study expected impact

- Offer practical guidelines for future APU-based deployments

Centre Informatique National de l’Enseignement Supérieur

HPC

IT Hosting

Archiving

CiNES

MINISTÈRE
DE L’ENSEIGNEMENT
SUPÉRIEUR,
DE LA RECHERCHE

France Universités

GENCI
Le calcul intensif au service de la connaissance

Hewlett Packard
Enterprise

Thank you

Questions?

Gabriel Hautreux (CINES)	Jean-Yves Vet (HPE)
Head of HPC Dep.	Application Performance Engineer
hautreux@cines.fr	vet@hpe.com

Unfruitful Strategies

Optimizations Explored Included:

- Changing memory frequency
- Switching CPU mode in the BIOS from *Power* to *Performance*
- Making PCIe links renegotiate to PCIe 4.0, 3.0, 3.0 or 2.0
- Use of other broadcast algorithms *Ibcast (=6)* increased performance on single node only

Software Environment

System

RHEL 9.4

Slurm 23.02.6

Slingshot Host Software (SHS) 11.0 providing libfabric 1.20.1

ROCm 6.2.0

User Space

rocHPL 6.0 (commit a394f17)

GCC 13.2

BLIS 4.2

rocBLAS 4.2.0

Cray MPICH 8.1.30

Slurm 23.02.6