“Characterizing GPU Energy Usage in Exascale-Ready Portable Science Applications”

CONNECTING THE DOTS

ISC High Performance

JUNE 10 – 13, 2025 | HAMBURG, GERMANY

Prepared for: The EESP Workshop 2025
Energy Efficiency with Sustainable Performance: Techniques, Tools, and Best Practices

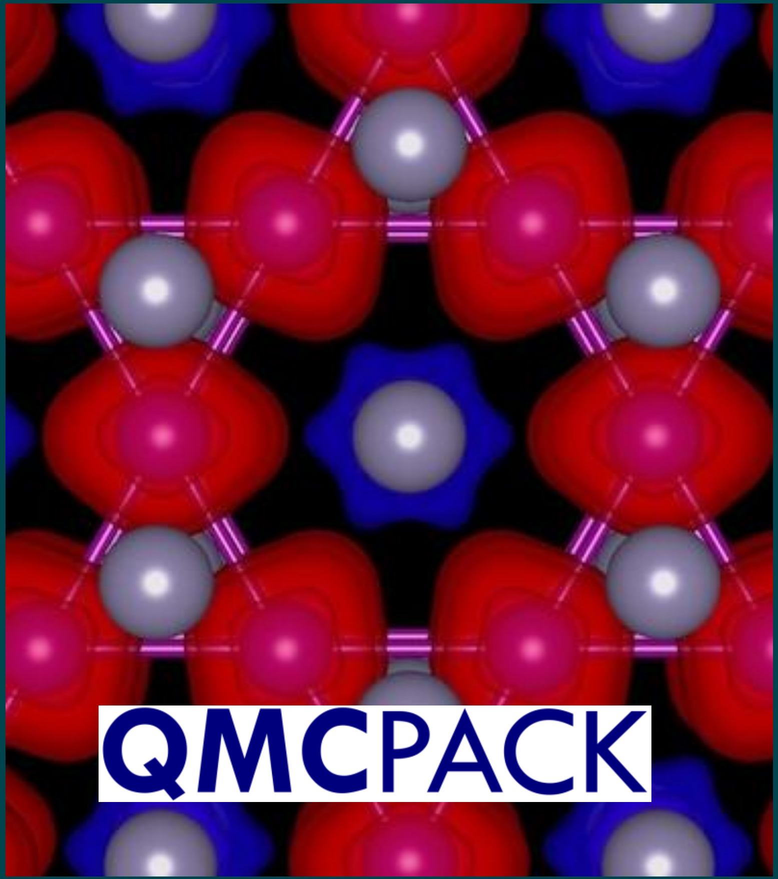
EESP WORKSHOP 2025

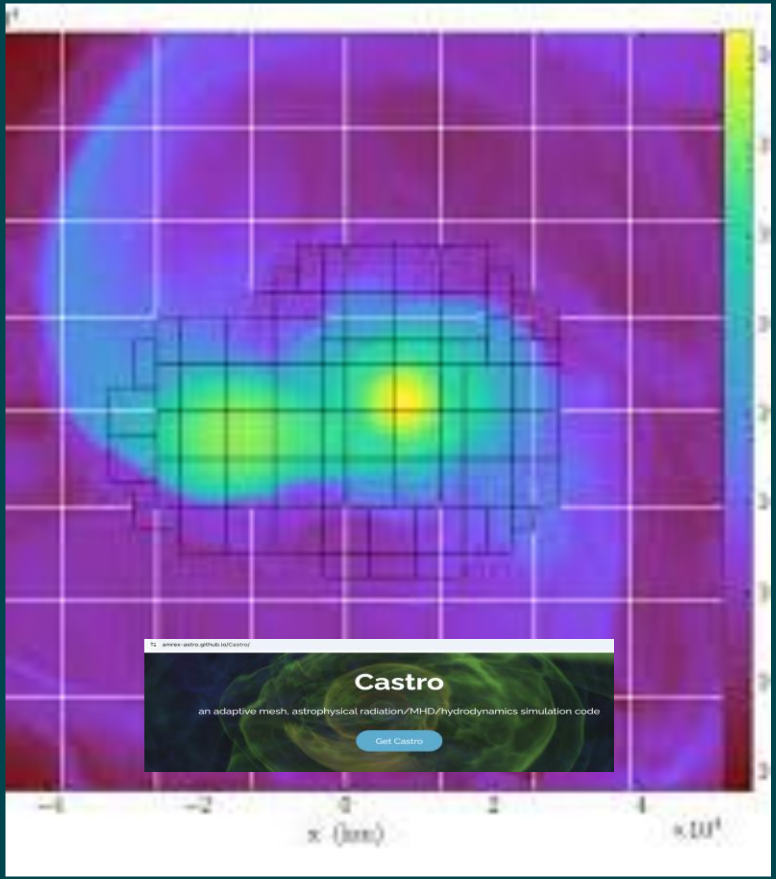
William F. Godoy[1], Oscar Hernandez[1], Paul R. C. Kent[1], **Maria Patrou**[1*], Kazi Asifuzzaman[1], Narasinga Rao Miniskar[1], Pedro Valero-Lara[1], Jeffrey S. Vetter[1], Matthew D. Sinclair[2], Jason Lowe-Power[3], Bobby R. Bruce[3]

[1] Oak Ridge National Laboratory, Oak Ridge, Tennessee, USA * Presenter
[2] University of Wisconsin-Madison, Madison, Wisconsin, USA
[3] University of California Davis, Davis, California, USA

OAK RIDGE National Laboratory
WISCONSIN UNIVERSITY OF WISCONSIN-MADISON
UCDAVIS COMPUTER SCIENCE

Overview

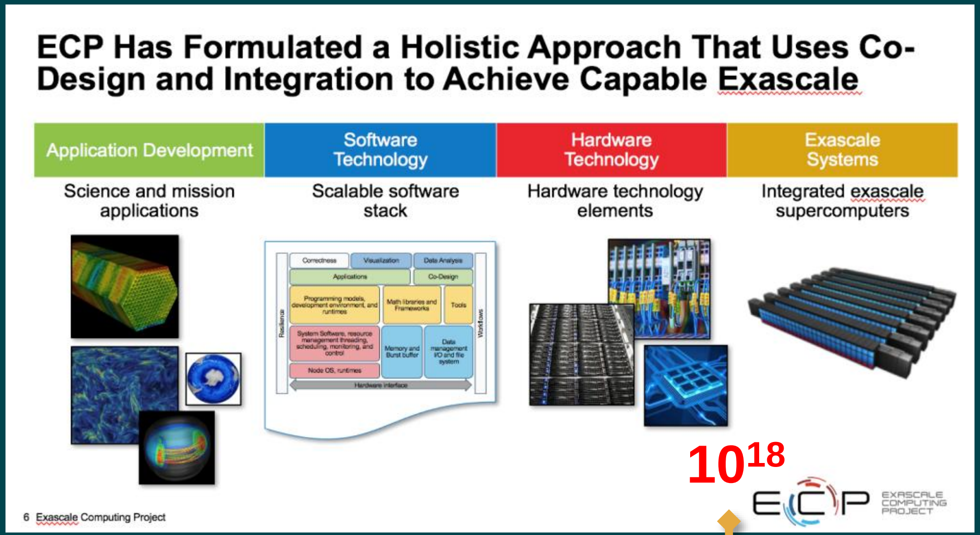


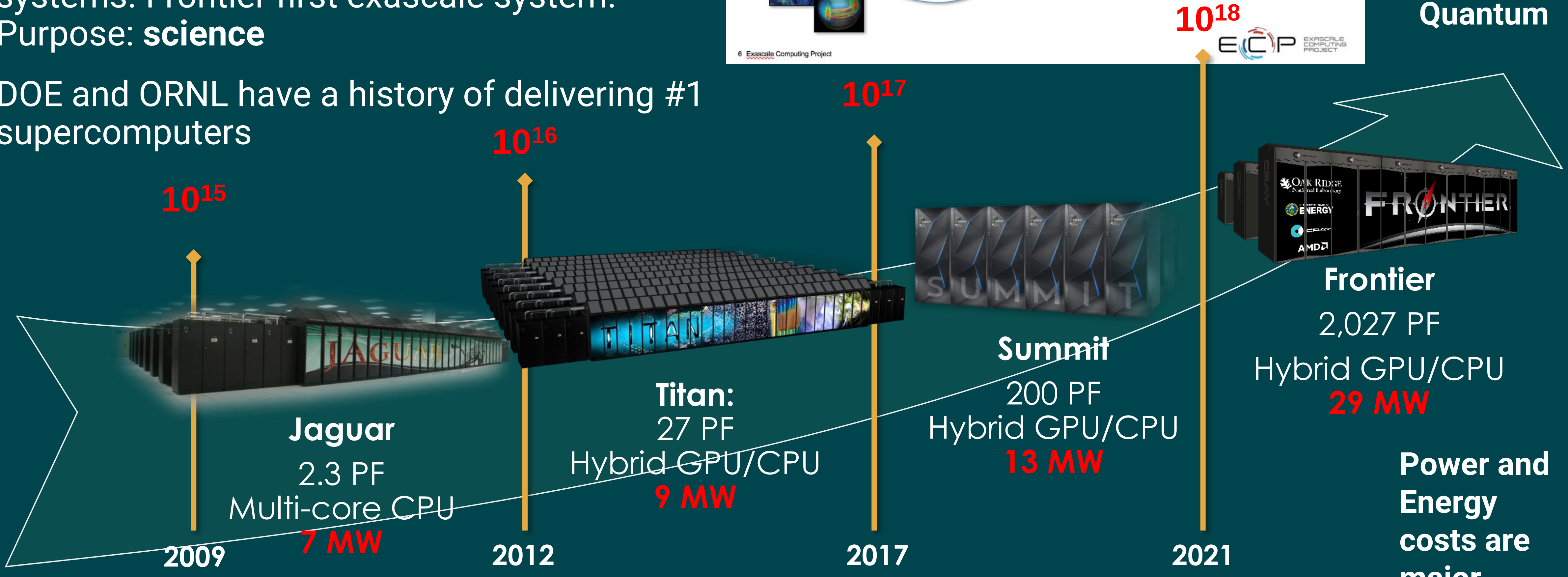


- Motivation
- Exascale applications: QMCPACK and AMReX
- Methodology: HWEnergyTracer.jl
- Results
- Conclusions and Future Work

Motivation

- US Department of Energy funded the Exascale Computing Project (ECP 2016-2023) 7 years $1.8B
- Deployed exascale computing on GPU systems. Frontier first exascale system. Purpose: **science**
- DOE and ORNL have a history of delivering #1 supercomputers



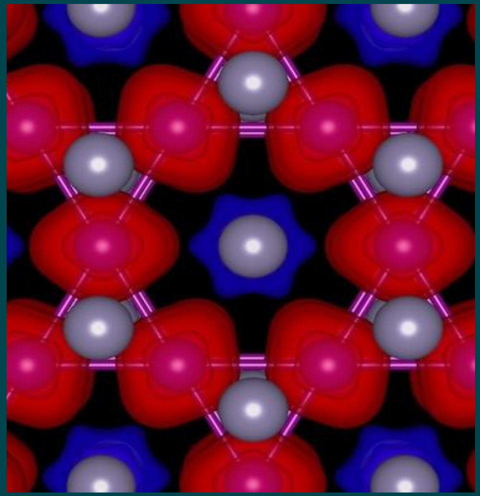


Power and Energy costs are major barriers

Exascale Applications: QMCPACK and AMReX-Castro

QMCPACK: particle

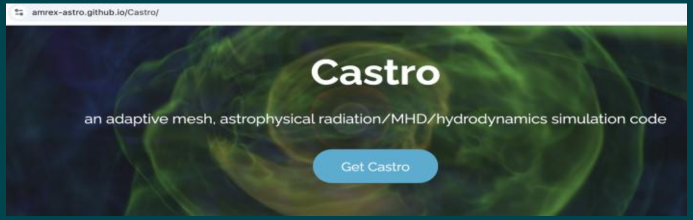
- Quantum Monte Carlo “ab-initio” code for 2D electrons, atoms, solid structures
- Developed since the the early 2000s. Ported to a new GPU methodology during ECP
- C++17, OpenMP offload, supports mixed-precision
- The Nickel-Oxide (NiO) benchmark is used for OLCF-6 (future system) procurement

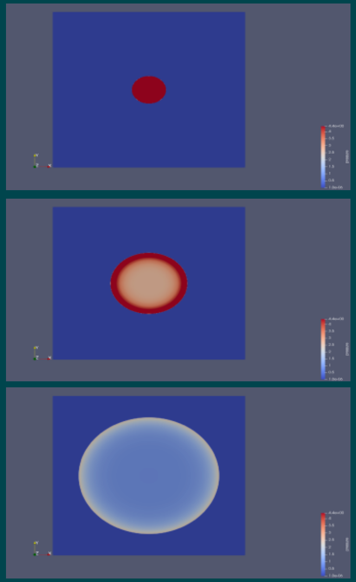


https://www.olcf.ornl.gov/wp-content/uploads/OLCF-6_QMCPACK_description-1.pdf

AMReX-Castro: adaptive mesh

- Astrophysical radiation/MHD/hydrodynamics simulation code based on AMReX
- Developed since early 2010s. Ported to GPU and improved during ECP
- Fortran, C++, vendor GPU: CUDA, HIP, SYCL, support for double and single-precision
- Sedov blast wave case





Methodology

Research questions:

- **Can we characterize GPU energy usage in a portable way? Target: NVIDIA and AMD GPUs**
- **What is the impact of mixed-precision on application energy consumption?**
- **What are the metrics to express "science-per-energy/power unit"?**

Steps:

- **Run vendor query tools (NVML, rocm_smi_lib) to capture power, GPU utilization, temperature and calculate energy of application benchmarks**
- **Understand impact of (i) time resolution: 1 ms to 1s, (ii) mixed-precision on measurements**
- **Propose unifying and comparable metrics (QMCPACK only, AMReX TBD)**

Table 1: Queries used for NVIDIA's NVML and AMD's rocm_smi_lib

Metric	**Relevant Query**	**Description**
NVIDIA	**nvmlDeviceGet***	
Power (W)	PowerUsage	Power usage of the GPU and its associated circuitry (e.g., memory) averaged over a 1 s interval [34]
Utilization (%)	UtilizationRates	Percent of time over the past sample period, between 1 and $\frac{1}{6}$ s, during which kernels were executing
Temperature (°C)	Temperature	Current temperature readings for the device
AMD	**rsmi_dev_***	
Power (W)	power_ave_get	device energy counter average for a short time (1 ms)
Utilization (%)	busy_percent_get	Percentage of time busy processing
Temperature (°C)	temp_metric_get	Retrieved from the temperature sensor for the device

Table 2: System hardware and software used in this study

System	**Milan0**	**Hudson**	**Frontier**
Hardware			
GPU-per-node	2 NVIDIA A100	2 NVIDIA H100	8 GCD AMD MI250X
Memory(GB)/Bandwidth(GB/s)	HBM2E 80/1,940	HBM3 94/1,940	HBM2E 64/3,276
Thermal Design Power (W)	300	400	500
Software			
GPU Tool Chain	NVHPC 24.9	NVHPC 24.9	ROCm 6.2
QMCPACK	v3.17.1	v3.17.1	v3.17.1
Compiler	Clang 19.1	Clang 19.1	AMDClang 6.2
Programming Model	OpenMP-offload	OpenMP-offload	OpenMP-offload
AMReX-Castro	v24.12	v24.12	v24.12
Compiler	GCC 13.2	GCC 13.2	GCC 12.3
Programming Model	CUDA 12.4	CUDA 12.4	HIP 6.2

Methodology: HWEnergyTracer.jl artifact tool

- **Portable Julia NVML wrapper** https://github.com/JuliaORNL/HWEnergyTracer.jl
- **Why Julia? Lightweight foreign function interfaces (FFIs) for GPU back ends: e.g. NVML is in inside CUDA.jl and rocm_smi_lib C API**
- **Start: 15 second before the app: (i) Julia JIT, (ii) static power state**

Start

```
echo "Starting HWEnergyTracer.jl"
julia -t 1 --project=$HWTracer_DIR $HWTracer_DIR/hw-energy-tracer.jl -v NVIDIA -
r 10 -o power_NiO-S128_w$walkers-production.csv > power.log 2>&1 &
sleep 15
```

App

```
echo "Starting QMCPACK"
$QMCPACK_EXE $xml_input > run_w$walkers.log 2>&1 &
qmcpack_pid=$!
```

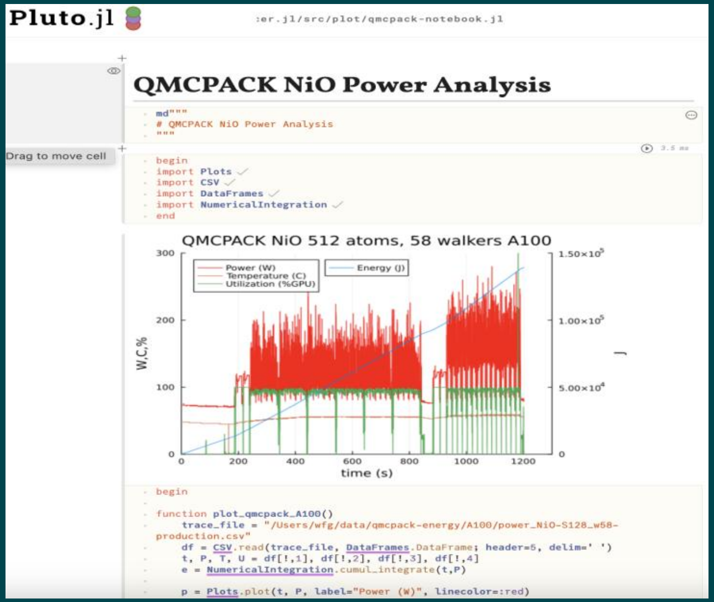
End

```
wait "$qmcpack_pid"
echo "End QMCPACK"
sleep 10
kill -2 "$power_pid"
echo "End HWEnergyTracer.jl"
```

CSV output: power, temperature, GPU utilization

```
$ head power_NiO-S128_w68-production.csv
NVIDIA NVML Power Trace
device_id 0
sample_rate 10.0
total_en ergy
Time(s)          Power(W)  Temperature(C) Util.gpu(%) Util.mem(%)
0.000000 61 38 0 0
0.262113 61 38 0 0
0.273473 61 38 0 0
```

Interactive data analysis



Results: QMCPACK NiO Benchmark

https://www.olcf.ornl.gov/wp-content/uploads/OLCF-6_QMCPACK_description-1.pdf

QMCPACK

Characteristics

- 2 main regions of interest:
  (i) Variational Monte Carlo (VMC),
  (ii) Diffusion Monte Carlo (DMC, largest use of time in production runs)
- DMC: 4 GPU streams launched to maximize GPU usage with several kernel launches/stream
- Well defined steps/blocks regions
- Each ”walker” is an independent Markov chain. More walkers = more accurate results.

(a) 5 steps x 3 blocks = 15 iterations (b) 1 ms snapshot

Fig. 1: QMCPACK NiO Benchmark DMC GPU traces on an NVIDIA H100.

Results: QMCPACK NiO Benchmark time resolution on NVIDIA

Observations:

- at 1 ms we see false positives
- at 1 s, nvidia-smi misses some GPU utilization patterns in DMC
- H100 results shows smoother traces aligning with the kernel characteristics
- A100 results are more noisy

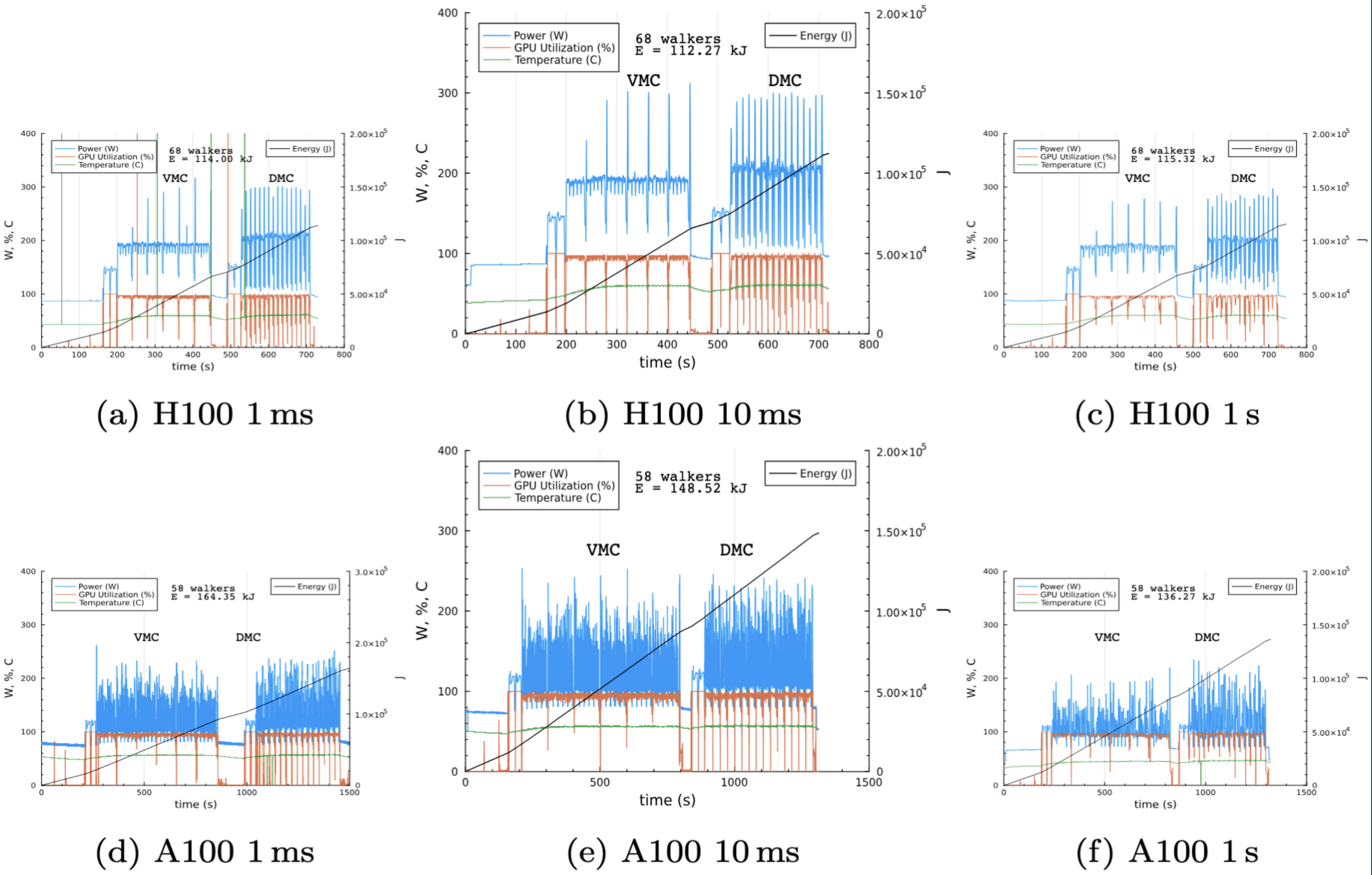


Fig. 2: Energy characteristics of the QMCPACK NiO benchmark on NVIDIA H100 and A100 for different query time resolutions.

OAK RIDGE National Laboratory

Results: QMCPACK NiO Benchmark time resolution on AMD

Observations:

- Measured on Frontier MI250X using rocm_smi_lib
- Resolution is more unclear. GPU utilization goes from 0 to 100 quickly. We stay with 1 ms for future runs
- We see a power drop in DMC as opposed to NVIDIA GPUs.
- Power consumption is far less optimal than A100
- Need for better tooling on the AMD side

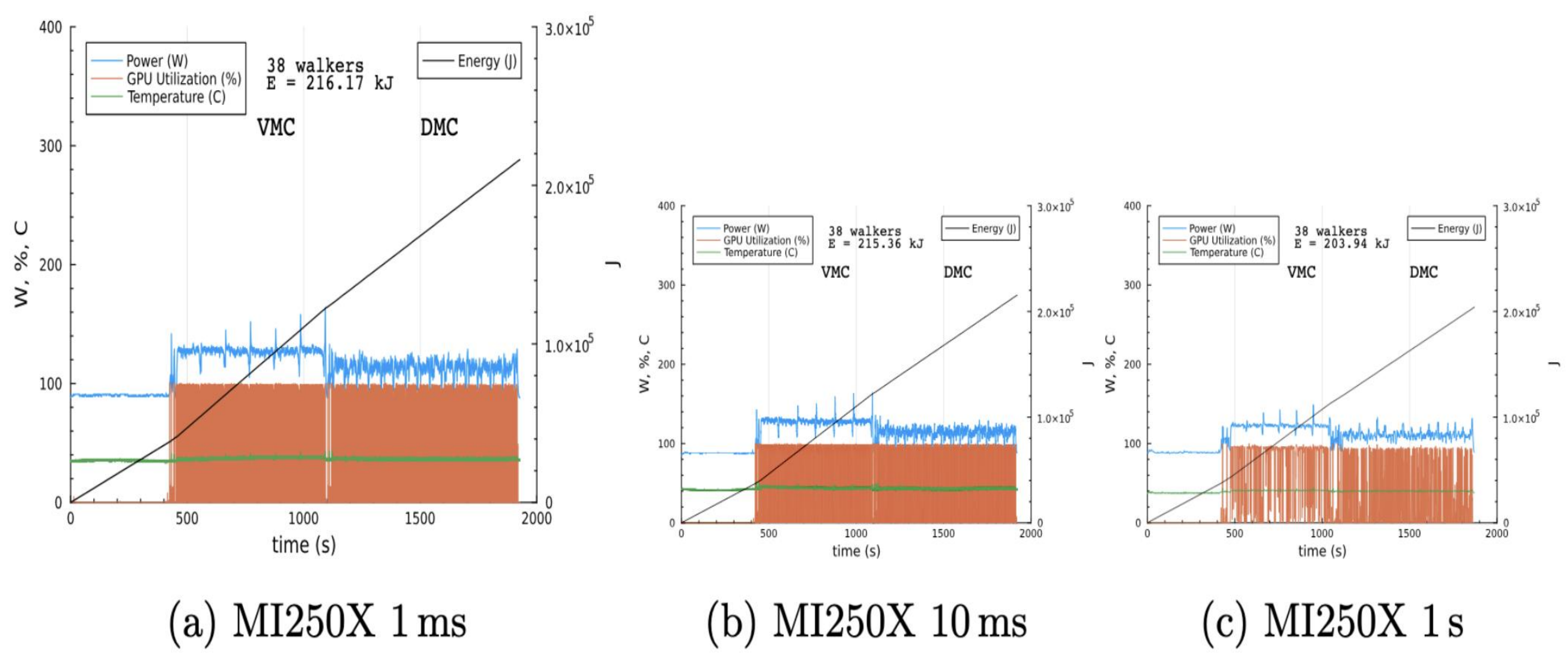
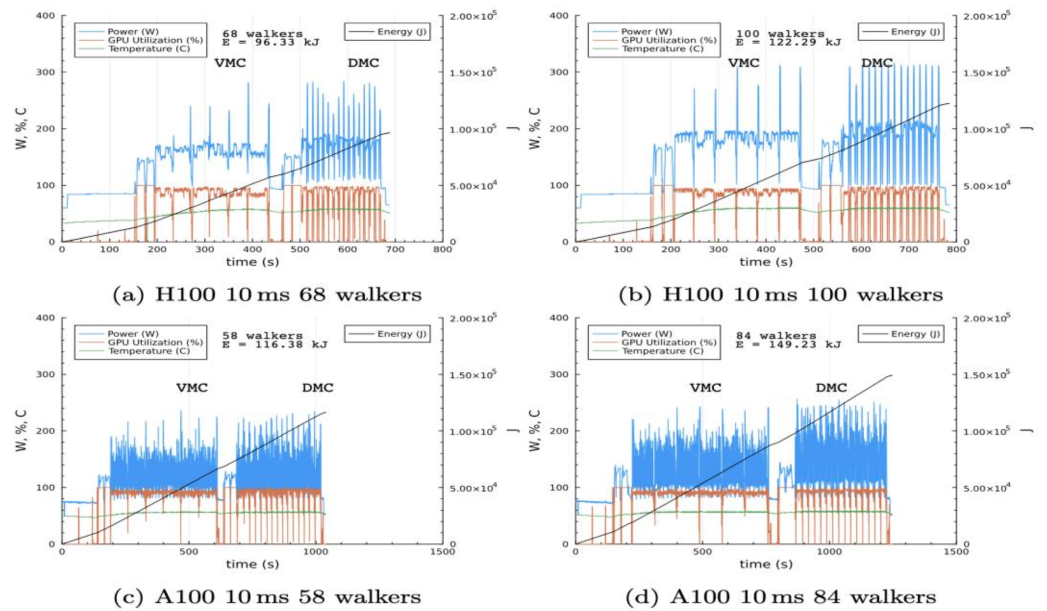


Fig. 3: Energy characteristics of the QMCPACK NiO benchmark on an AMD MI250X for different query time resolutions.

Results: QMCPACK NiO Benchmark mixed-precision NVIDIA

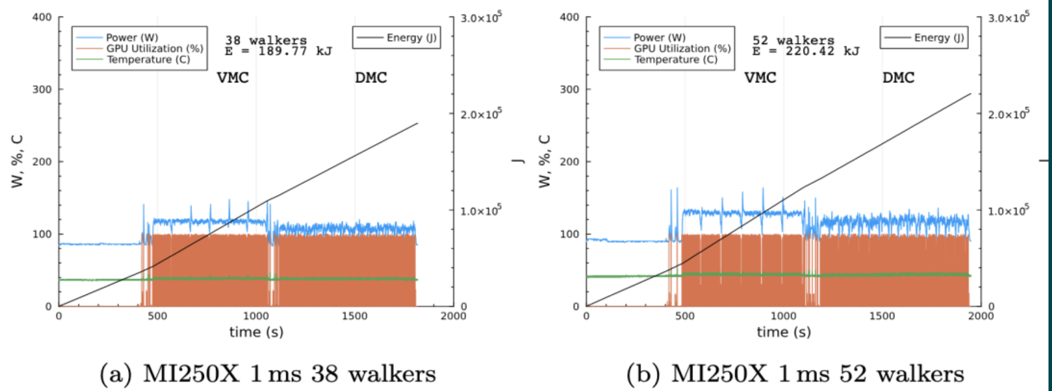
Observations:

- Two ways to use reduced “mixed-precision”: (i) maximize performance (same walkers e.g. 68), (ii) maximize science (more walkers, e.g. 100)
- In a real application like QMCPACK some components can’t use reduced precision
- Energy savings are in the range of 6-25% for mixed-precision for the same number of walkers
- Energy savings due to shorter time-to-solution, power utilization is similar
- Bonus: power/energy analysis identified different default behaviors in the code: https://github.com/QMCPACK/qmcpack/pull/5248



(a) H100 10 ms 68 walkers (b) H100 10 ms 100 walkers (c) A100 10 ms 58 walkers (d) A100 10 ms 84 walkers

Fig. 4: Mixed-precision traces on NVIDIA H100 and A100 for (a) max double-precision walkers (68 and 58) and (b) max mixed-precision walkers (100 and 84).



(a) MI250X 1 ms 38 walkers (b) MI250X 1 ms 52 walkers

Fig. 5: AMD MI250X mixed-precision traces for (a) max double-precision walkers (38) and (b) max mixed-precision walkers (52).

OAK RIDGE National Laboratory

Results: QMCPACK NiO Benchmark "Energy Metric"

Metric

- Science/Energy: Walkers moved/kJ

$$\texttt{Throughput}_{\texttt{Energy}} = \frac{\texttt{walkers} \times \texttt{blocks} \times \texttt{steps}}{\texttt{DMC energy}},$$

- Only taken from the DMC region of interest
- Uniform metric to compare across hardware and future-proof for energy efficiency
- Time-to-solution implicitly included in the denominator
- NVIDIA H100+mixed precision is the most energy efficient despite lower GPU utilization %

Table 3: Energy metrics for QMCPACK's DMC on the NiO a512 benchmark for multiple GPU configurations

Configuration	Walkers *Max	Throughput Energy (1/kJ)	Power (W)	GPU (%)
NVIDIA				
H100-mixed	*100	38.69	190.02	72.54
H100-mixed	*68	33.20	172.60	74.60
H100-double	*68	27.26	191.87	83.92
H100-mixed	84	37.56	182.31	74.69
H100-mixed	58	32.42	174.24	75.93
H100-double	58	26.28	181.89	78.58
A100-mixed	*84	25.25	136.10	85.76
A100-mixed	*58	20.86	121.59	85.08
A100-double	*58	17.31	124.97	88.03
A100-mixed	52	21.41	122.22	83.84
A100-mixed	38	16.13	109.01	86.92
A100-double	38	15.58	119.11	86.69
AMD				
MI250X-mixed	*52	9.12	115.97	39.33
MI250X-mixed	*38	7.57	106.36	40.12
MI250X-double	*38	6.19	112.61	39.27

Results: AMReX-Castro Sedov blast case

Characteristics

- Adaptive Mesh Refinement advances cells based on the problem characteristics (e.g. Courant–Friedrichs–Lewy condition)
- Sedov blast is a simple explosion benchmark problem with analytical solutions
- 4 mesh levels evolution dictates the energy characteristics. L3 – the smallest resolution -- being the dominant driver

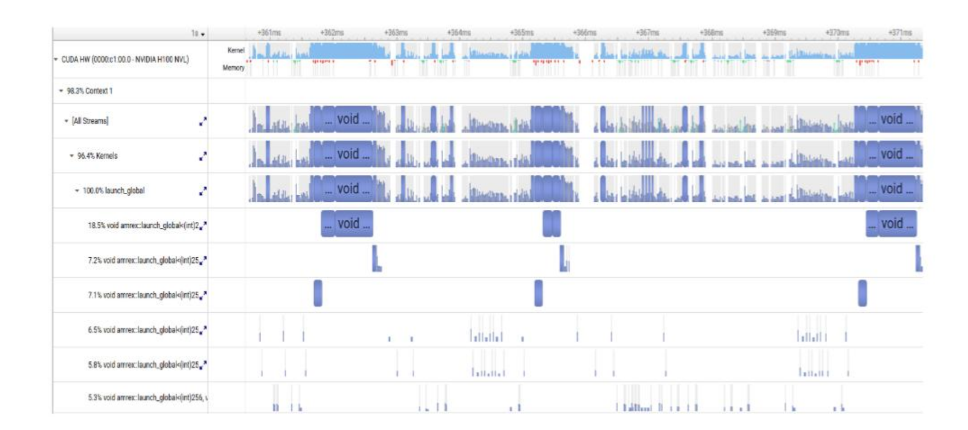
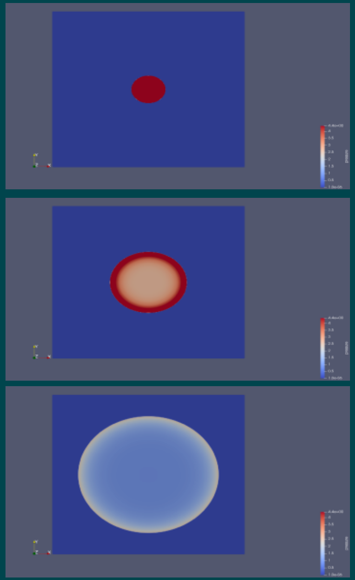


Fig. 9: AMReX-Castro Sedov simulation 10 ms trace snapshot on an NVIDIA H100 using a 256 × 256 base mesh.



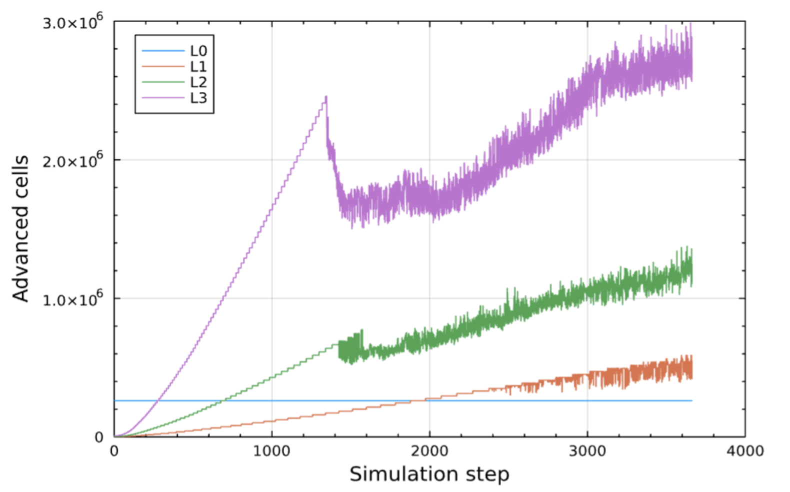
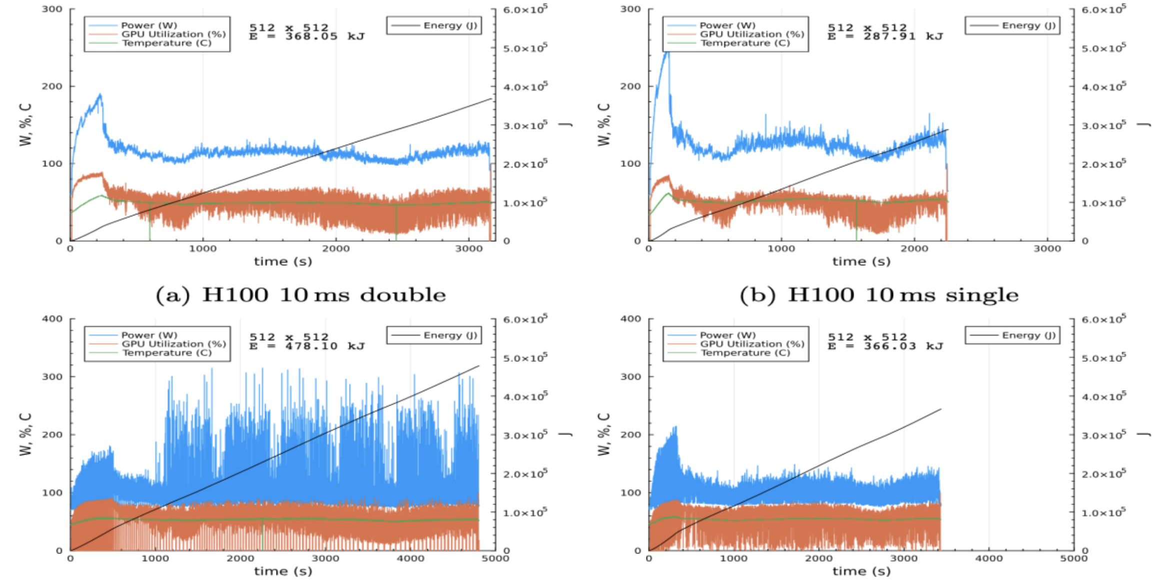


Fig. 10: AMReX-Castro Sedov problem advanced cells per AMR mesh level as a function of simulation step.

Results: AMReX-Castro Sedov Case on NVIDIA and AMD

Observations:

- Single-precision achieves energy savings over double precision.
- A100 has larger power peaks and variability (tool characteristic?)
- We could not find differences between single and double-precision on MI250X
- Energy characteristics largely driven by the smallest resolution mesh
- A unifying science-per-watt metric is still TBD (which mesh size? average? maximum?).



(a) H100 10 ms double (b) H100 10 ms single

(c) A100 10 ms double (d) A100 10 ms single

Fig. 7: AMReX-Castro Sedov energy characteristics on NVIDIA H100 and A100 for a 512 × 512 base mesh with CFL = 0.25 using (a) double and (b) single precision.

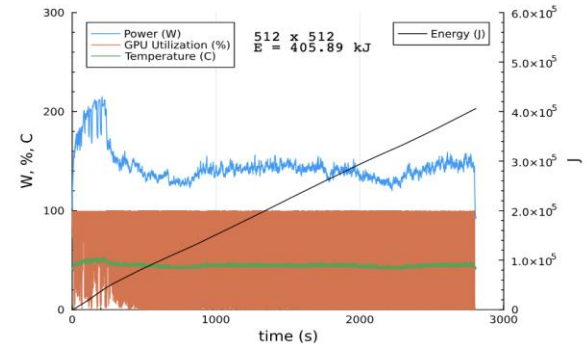


Fig. 8: AMReX-Castro Sedov energy characteristics on an AMD MI250X for a 512 × 512 base mesh using double precision, 1 ms resolution, and CFL = 0.25.

Final remarks

Summary:

- We analyze two widely-used applications: (i) QMCPACK and (ii) AMReX-Castro on NVIDIA and AMD GPUs including effects of different precision levels and a discussion on science-per-energy unit metrics
- **Observation 1:** small variability for the power, utilization, temperature traces on NVML (10 ms) and rocm smi lib (1 ms). Energy tracing can be easily integrated in the application's software development process
- **Observation 2:** reduced floating-point precision energy-savings 6%-25% on QMCPACK (on NVIDIA and AMD) and AMReX-Castro (on NVIDIA only)
- **Observation 3:** Energy-efficiency improvements (on the order of 1.5×) were shown for NVIDIA's H100 over their A100. Room for improvement exists for AMD's GPU tools and applications as the ecosystem matures.
- **Observation 4:** proposed QMCPACK metric shows science-per-energy unit, AMR codes is still TBD due to mesh variability
- Future work: extend to more applications, hardware, and science-per-energy metrics

Acknowledgements

This material is based on work supported by the DOE's Office of Science, Office of Advanced Scientific Computing Research through EXPRESS: 2023 Exploratory Research for Extreme Scale Science.

PRCK was supported by the DOE's Office of Science, Basic Energy Sciences, Materials Sciences and Engineering Division as part of the Computational Materials Sciences Program and the Center for Predictive Simulation of Functional Materials.

This research used resources of the Oak Ridge Leadership Computing Facility and the Experimental Computing Laboratory at the Oak Ridge National Laboratory, which is supported by the DOE's Office of Science under Contract No. DE-AC05-00OR22725.

We would like to acknowledge Brandon Tran from the University of Wisconsin for the valuable discussion on NVML.

Thank you!
Questions?

Oak Ridge National Laboratory