# Power-Centric Observability for HPC Systems: Design, Deployment, and Evaluation on REPACSS


Yongjian Zhao
Department of Computer Science
Texas Tech University
Lubbock, Texas, USA
yongjian.zhao@ttu.edu

Jie Li
Department of Computer Science
Texas Tech University
Lubbock, Texas, USA
jie.li@ttu.edu

Chenxu Niu
NVIDIA Corporation
Santa Clara, California, USA
chenxun@nvidia.com

Alan Sill
High Performance Computing Center
Texas Tech University
Lubbock, Texas, USA
Alan.Sill@ttu.edu

Yong Chen
Department of Computer Science
Texas Tech University
Lubbock, Texas, USA
yong.chen@ttu.edu



## Abstract

Energy efficiency is a critical concern in modern heterogeneous HPC systems, where CPUs, GPUs, and dense cooling infrastructure increase both idle and peak energy demand. This paper presents a power-centric observability framework deployed on the NSF REPACSS system that enables non-intrusive and consistent energy analysis across facility, rack, node, and job scopes. The framework collects instantaneous power telemetry exclusively from out-of-band hardware sources, including in-row cooling units, rack power distribution units, and compute nodes accessed via iDRAC. These measurements are stored in a time series database and processed through a power-centric energy model that aligns heterogeneous telemetry in time and space. A scalable query engine and web API provide a uniform interface for deriving energy metrics across infrastructure and job contexts, while a lightweight Slurm epilog integration enables optional job level energy reporting without application modification or runtime instrumentation. Using this unified telemetry and query framework, we conduct a multi-scope energy characterization of the NSF REPACSS data center. The results show stable facility efficiency, efficient in-row cooling behavior, distinct differences in idle and peak power between CPU and GPU nodes, and diverse non-CPU energy contributions in representative workloads. Together, these findings demonstrate that unified out-of-band telemetry combined with a power-centric energy model, enables practical cross-scope energy observability in production HPC systems.



## CCS Concepts

• **Computer systems organization → Heterogeneous (hybrid) systems**; • **Hardware → Power and energy**.




ACM ISBN 979-8-4007-2377-3/2026/07
https://doi.org/10.1145/3785462.3815795



## Keywords

HPC Systems, Power Observability, Out-of-band Telemetry, Slurm Integration

**ACM Reference Format:**
Yongjian Zhao, Jie Li, Chenxu Niu, Alan Sill, and Yong Chen. 2026. Power-Centric Observability for HPC Systems: Design, Deployment, and Evaluation on REPACSS. In *Practice and Experience in Advanced Research Computing (PEARC '26), July 26–30, 2026, Minneapolis, MN, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3785462.3815795


## 1 Introduction

Modern high-performance computing (HPC) systems increasingly rely on heterogeneous compute resources, including multi-core CPUs, high-power GPUs, and specialized accelerators to support scientific simulation, AI workloads, and data-intensive applications. Although these resources deliver high throughput, they also introduce significant energy challenges, including high idle power consumption, increased thermal density, and growing cooling and power delivery overheads at the node, rack, and facility scales. As a result, understanding and optimizing the energy behavior throughout the computing system has become an important requirement for sustainable and cost-effective HPC operation.

Currently data center efficiency is commonly measured through Power Usage Effectiveness (PUE), which compares total facility energy consumption with the energy used by IT equipment. While PUE provides a useful aggregate metric, it offers limited insight into where energy is consumed within the system, how cooling and power infrastructure respond to workload dynamics, or how individual jobs contribute to overall energy use. In production environments, system operators often face the "siloed data" challenge, where power telemetry is fragmented across disjoint sources such as rack PDUs, cooling controllers, and node level management interfaces, making cross-layer correlation manually intensive and prone to error. These limitations become more pronounced in heterogeneous systems, where CPU and GPU nodes exhibit drastically different thermal footprints.

At the same time, both system operators and users increasingly require fine-grained energy visibility to inform job scheduling and performance optimization. Prior work has shown that energy-aware system optimization benefits from telemetry spanning facility, rack, node, and application levels [11]. In practice, however, such visibility remains difficult to achieve without intrusive instrumentation that may impact application performance or violate security policies in shared research environments. Furthermore, users typically lack direct access to energy information associated with their specific jobs, preventing the community from adopting energy-efficient programming practices.

This paper presents a *power-centric observability framework* deployed on the NSF-funded REPACSS (**RE**motely-managed **P**ower **A**ware **C**omputer **S**ystems and **S**ervices) data center at Texas Tech University. Our framework bridges the gap between facility operations and user level accounting by collecting instantaneous power telemetry from out-of-band (OOB) hardware layers, including in-row cooling (IRC) units, rack PDUs as well as compute nodes, and composing these signals into unified energy views. By focusing on OOB telemetry, we provide a non-intrusive solution that remains persistent across OS reboots and application execution.

Our contributions are summarized as follows:

(1) **Unified multi-source power telemetry collection.** We design and deploy a resilient metric collection pipeline that gathers continuous power data from heterogeneous sources, including in-row cooling (IRC) units, rack level PDUs, and compute nodes via Dell iDRAC. Measurements are collected using standard management protocols (SNMP and Redfish) and stored in a time series database for persistent, system-wide visibility.
(2) **A power-centric energy model with scalable query support.** We introduce a power-centric energy model that aligns heterogeneous telemetry across nodes, racks, and cooling zones, and derives consistent energy views through time alignment, aggregation, and attribution. The model is backed by a scalable query engine and Web API, decoupling analysis consumers from raw sensor interfaces and enabling flexible energy queries across multiple system scopes.
(3) **Lightweight job level energy reporting via Slurm integration.** We integrate the observability framework with the Slurm workload manager using an epilog-based mechanism that enables optional job level energy reporting without application modification or scheduler intrusion. This design exposes job-aligned energy information to users while remaining compatible with production HPC workflows.
(4) **Cross-scope energy analysis enabled by unified telemetry.** Using the proposed framework, we demonstrate how a unified out-of-band telemetry foundation enables consistent energy analysis across multiple scopes, including facility level PUE characterization, rack level cooling behavior, node level CPU and GPU power profiling, and application level component-wise energy breakdowns.

Together, these contributions provide a practical foundation for power-centric, cross-layer power observability in heterogeneous HPC systems. By integrating out-of-band telemetry with a unified query interface and Slurm based job reporting, the framework enables repeatable energy attribution from infrastructure to user jobs and supports energy-aware optimization in production environments.

# 2 Background

Power monitoring is a fundamental capability in modern HPC systems, supporting performance tuning, operational efficiency, cost control, and energy-aware scheduling. As HPC clusters grow in scale and heterogeneity, especially with the adoption of high power accelerators and dense cooling infrastructure, accurate and reliable power telemetry becomes increasingly important for both system administrators and users.

This section introduces the physical organization of the NSF REPACSS data center, and the out-of-band telemetry context on which the proposed power observability framework is built.

## 2.1 REPACSS Physical Layout Overview

NSF REPACSS data center is a heterogeneous HPC system deployed at Texas Tech University to support compute-intensive scientific workloads and emerging AI applications. As shown in Figure 1, the system[1] spans seven racks (Rack 91–Rack 97) and consists of 110 CPU compute nodes, 8 GPU nodes (each with 4 NVIDIA H100 GPUs), storage systems, network switches, and supporting utility nodes interconnected via InfiniBand and Ethernet networks.

REPACSS integrates power telemetry from a limited set of out-of-band sources that cover three domains: cooling, rack level power delivery, and compute nodes. At the infrastructure level, in-row cooling (IRC) units expose compressor and fan power, while rack PDUs report total rack power. At the compute level, a subset of servers expose node input power and selected component-wise signals (such as CPU power, DRAM power, etc.) through iDRAC. Components highlighted in blue in Figure 1 indicate elements with active out-of-band telemetry.

## 2.2 Out-of-Band Power Telemetry

Power telemetry in HPC systems is commonly obtained using either in-band or out-of-band approaches. In-band monitoring relies on OS-level interfaces such as Intel RAPL [3] and NVIDIA NVML [19], but typically requires a running OS, elevated privileges, and integration in the software stack [15].

REPACSS primarily uses out-of-band (OOB) telemetry collected through embedded baseboard management controllers (BMCs). For compute nodes, Dell iDRAC provides node level power measurements via Redfish [2, 13]; for infrastructure equipment, facility devices expose rack and cooling power through the standard management interface SNMPv3. This choice supports persistent measurement and enables aligning telemetry with system operation and job execution.

Together, the organization of the REPACSS system and its OOB telemetry capabilities establish the measurement context for the framework in Section 3.

# 3 Power Observability Framework Design

This section presents the design of the REPACSS power observability framework, which provides end-to-end, power-centric visibility into energy behavior from infrastructure components to user level jobs. The framework is designed to be non-intrusive, scalable, and tightly integrated with existing HPC management software.

[1] https://www.repacss.org/resources/

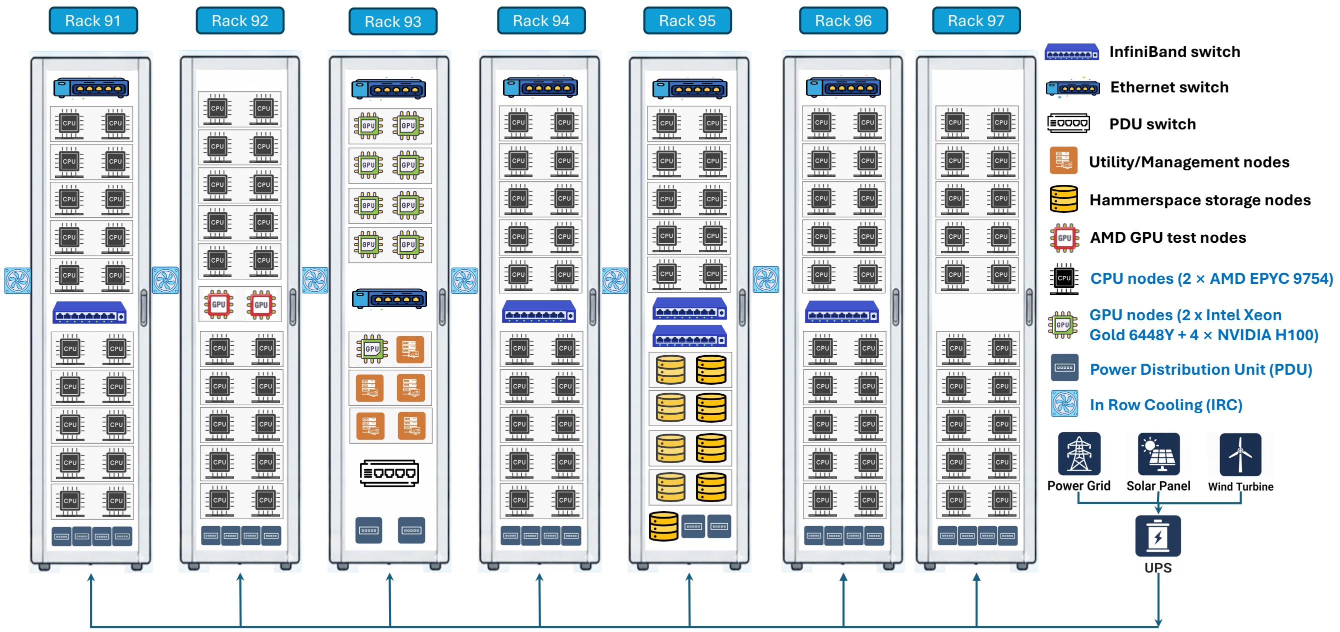


**Figure 1: Physical layout of the NSF REPACSS data center across seven racks. Components highlighted in blue indicate elements with active out-of-band power telemetry.**

## 3.1 Overview

Figure 2 illustrates the end-to-end power observability framework deployed on REPACSS. The framework follows a power-centric, telemetry-driven workflow in which power measurements are collected from cooling infrastructure, rack power delivery, and compute nodes, processed by a centralized observability core, and exposed as energy views at the facility, rack, node, and job levels.

On the left side of the architecture, a limited set of out-of-band hardware sources continuously provide power measurements from in-row cooling units, rack-level power distribution units, and instrumented compute nodes. These sources operate independently of the host operating system and running workloads, enabling non-intrusive and continuously available data collection.

The collected telemetry is ingested into a centralized power observability core, where raw measurements are stored, normalized, and organized according to their physical context. Within this core, power signals from heterogeneous sources are composed into a unified representation that supports aggregation across spatial scopes and integration over time to derive energy metrics.

On the right side of the architecture, energy information is exposed to users through lightweight integration with the Slurm workload manager. Job execution metadata is used to align power telemetry with application runtime intervals, enabling transparent energy accounting at the job level without modifying applications or interfering with the scheduler's critical path.

## 3.2 Out-of-Band Hardware Sensors

REPACSS collects power telemetry exclusively from out-of-band management interfaces. These telemetry sources contribute to the power observability framework and operate independently of the host operating system and running workloads.

Power telemetry is obtained from three classes of hardware components:

- **In-row cooling (IRC).** In-row cooling units expose electrical power signals associated with cooling operation, including compressor and fan power, enabling direct accounting of cooling related energy consumption at the rack level.
- **Rack power distribution units (PDUs).** Rack level PDUs report total rack power, capturing the aggregate electrical load of each rack, including components that do not expose fine-grained power telemetry such as network switches and auxiliary equipment.
- **Compute nodes (iDRAC).** For instrumented servers, node management controllers expose system input and output power and, when supported, component-wise power signals for CPUs, memory, fans, storage and GPUs.

For each compute node, the difference between system input power and system output power is used to estimate power supply conversion losses (PSU loss), capturing energy dissipated during AC–DC conversion and internal power delivery.

Table 1 summarizes the telemetry signals used in this study and their sampling intervals.

## 3.3 Power Observability Core

The power observability core transforms heterogeneous out-of-band (OOB) telemetry into a unified and queryable energy representation. As shown in Figure 2, it comprises a metric collector, a time-series database, a power-centric energy model, and a query

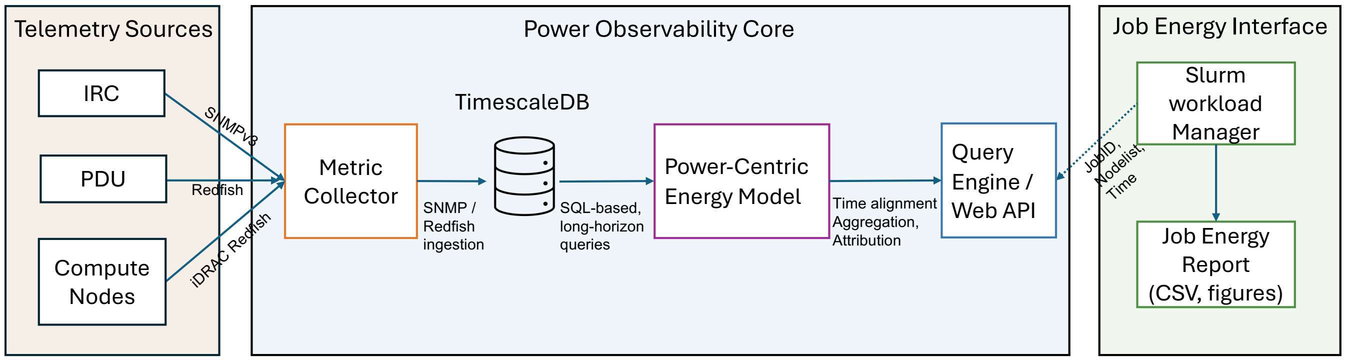


**Figure 2: End-to-end power observability framework deployed on REPACSS**

**Table 1: Hardware-level power telemetry sources used in REPACSS.**

| Telemetry Source | Metric | Units | Interval |
|---|---|---|---|
| PDU | PDUTotalPower | W | 60.0 s |
| IRC | CompressorPower | kW | 120.0 s |
| IRC | CondenserFanPower | W | 120.0 s |
| IRC | CoolDemand | kW | 120.0 s |
| Compute Node (iDRAC) | SystemInputPower | W | 5.0 s |
| Compute Node (iDRAC) | SystemOutputPower | W | 5.0 s |
| Compute Node (iDRAC) | TotalCPUPower | W | 5.0 s |
| Compute Node (iDRAC) | TotalMemoryPower | W | 5.0 s |
| Compute Node (iDRAC) | TotalStoragePower | W | 5.0 s |
| Compute Node (iDRAC) | TotalFanPower | W | 5.0 s |
| Compute Node (iDRAC) | PowerConsumption (per GPU) | mW | 1.0 s |

engine with a web API, and serves as the semantic bridge between hardware telemetry and higher-level energy analysis.

*Metric collector.* REPACSS continuously collects node and infrastructure telemetry from Redfish and SNMP endpoints. For compute nodes, the collector leverages the Redfish `TelemetryService` interface and subscribes to device-specific telemetry streams. Conceptually, the collector discovers telemetry capabilities on each node, then registers the corresponding sources for continuous ingestion:

> `GET https://<node>/redfish/v1/TelemetryService`
> → discover telemetry channels → register sources for push/stream ingestion.

This design avoids host-level instrumentation and provides a uniform collection path across heterogeneous nodes. Before storage, lightweight preprocessing is performed, including metric identifier normalization, unit handling, and mapping each sample to its physical context (node, rack, or cooling zone).

*Time-series storage (TimescaleDB).* All telemetry is stored in TimescaleDB to support long-horizon, SQL-native analytics over high-volume measurements. Its relational model simplifies joins with system metadata and enables efficient ad hoc queries for debugging, reporting, and offline analysis.

*Power-centric energy model.* The energy model encodes relationships among hardware entities and defines preprocessing rules required for cross-source analysis. Because sensors operate at heterogeneous sampling intervals, the model performs time alignment, missing sample handling, and energy integration over arbitrary intervals. These transformations enable higher level analyses used throughout this work, including PUE derivation, rack level power accounting, node level component breakdowns, and job-aligned energy attribution.

*Query engine and web API (FastAPI).* The query engine exposes the energy model through a Web API implemented with FastAPI. It supports queries across system, rack, and node scopes, as well as instantaneous power, interval-averaged power, and integrated energy metrics. By abstracting access to raw SNMP and Redfish endpoints and avoiding direct database interaction, the observability core provides a stable and uniform interface for administrator monitoring and user-facing job energy analysis.

## 3.4 Job Level Energy Reporting via Slurm

REPACSS exposes energy information to users through native integration with the Slurm workload manager using an epilog-based mechanism. The integration is implemented via `EpilogSlurmctld`[2] and executes outside the scheduler's critical path, requiring no custom Slurm plugins or application modifications.

Job level energy reporting is enabled in an optional manner. Users request energy profiling at submission time by attaching a comment to the job:

```
sbatch --comment=power_profiling {job_script}.sh
```

This design leverages an existing Slurm interface and preserves compatibility with standard job workflows.

Upon job completion, the epilog script captures job metadata, including the job ID, allocated nodes as well as execution time window and forwards these parameters to the power observability query API. The API returns the corresponding job-aligned energy results derived by the observability core.

The resulting job level energy report (Figure 3) provides time series power traces for major components and an aggregated energy decomposition across CPUs, memory, storage, fans, and power conversion losses. Reports are written to a user accessible location and referenced directly in the Slurm job output.

Currently, job-level energy attribution assumes exclusive node allocation, as the underlying iDRAC telemetry reflects whole node power consumption. Energy accounting for multiple jobs sharing

[2]https://slurm.schedmd.com/prolog_epilog.html

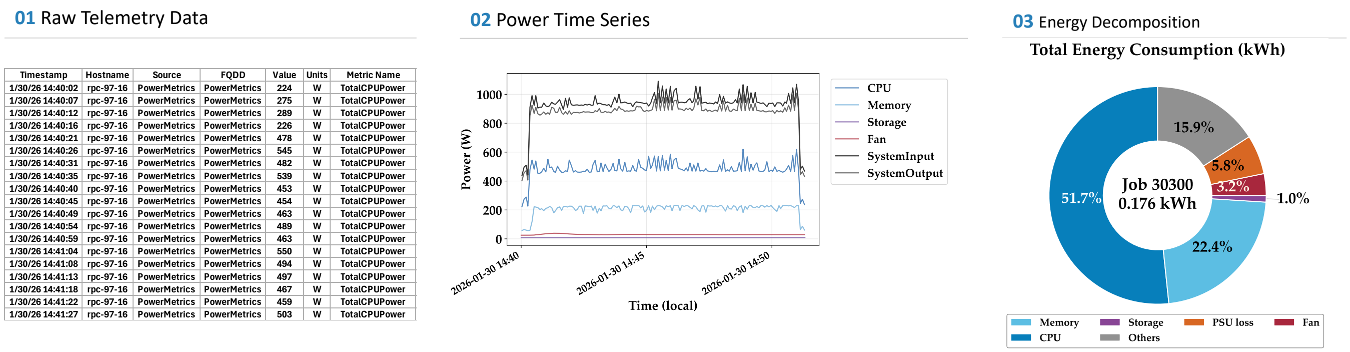


**Figure 3: Example job level energy report generated by REPACSS, showing time-series power traces and a component-level energy breakdown over the job execution interval**

a node is not supported in the current implementation. Extending the framework with power models to enable fine-grained energy attribution under node sharing is left as future work.

## 4 Experiments

This section evaluates the proposed power observability framework through multi-scope energy analyses conducted on the REPACSS system. Using the unified telemetry, power-centric energy model, and query interface described in Section 3, we construct energy views that capture both system level behavior and user facing workload characteristics.

Our evaluation covers four analysis levels: (i) facility level energy effectiveness through PUE derived from power delivery and cooling telemetry, (ii) rack level characterization of in-row cooling behavior, (iii) node level power profiling of CPU and GPU compute nodes, and (iv) application level energy analysis of representative HPC workloads using job-aligned telemetry.

Together, these experiments illustrate how a limited set of out-of-band power measurements can be composed into consistent energy views that support both system level understanding and energy-aware optimization from the user perspective.

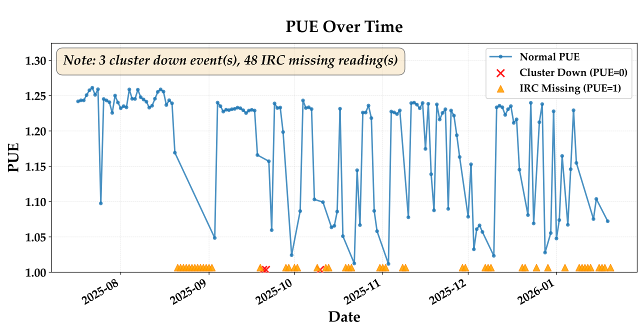


**Figure 4: PUE over time for REPACSS from initial operation through January 2026. Points corresponding to cluster downtime (PUE = 0) and missing or incomplete in-row cooling (IRC) telemetry (PUE = 1) are excluded from efficiency analysis.**

### 4.1 Facility Level Energy Effectiveness

Power Usage Effectiveness (PUE) is a standard metric for evaluating facility energy efficiency, defined as the ratio of total facility energy to IT equipment energy. In REPACSS, PUE is computed using aggregated rack level PDU energy as a proxy for IT load and in-row cooling (IRC) telemetry to account for cooling overhead.

Figure 4 shows the PUE of REPACSS from initial operation through January 2026. Under normal operation, PUE remains stable, with most values between 1.20 and 1.26. PUE = 0 indicates cluster downtime, while PUE = 1 reflects missing or partial IRC telemetry rather than ideal efficiency; these intervals are excluded from further analysis.

After filtering, REPACSS exhibits an effective PUE centered around 1.23 over its operational lifetime. This indicates that approximately 80–82% of facility energy is delivered to IT equipment, with the remaining 18–20% primarily attributable to in-row cooling. Short-duration PUE drops occur when IRC systems operate intermittently and do not represent steady-state behavior.

Overall, these results place REPACSS within the efficiency range reported for well-optimized enterprise and research HPC facilities, demonstrating that localized in-row cooling can support stable and energy-efficient operation in a heterogeneous production environment.

### 4.2 Rack Level Cooling Behavior

At the rack level, we evaluate how effectively in-row cooling (IRC) units convert electrical power into delivered cooling. This analysis connects facility level efficiency metrics PUE with localized thermal behavior observed at the rack scale.

The analysis uses the IRC telemetry signals summarized in Table 1. Specifically, *CompressorPower* and *CondenserFanPower* represent the electrical input to the cooling system, while *CoolDemand* serves as a proxy for delivered cooling output. All three signals are sampled at 120 s intervals and collected exclusively through out-of-band interfaces.

Cooling efficiency is quantified using the coefficient of performance (COP), defined as:

$$\text{COP} = \frac{\mathit{CoolDemand}}{\mathit{CompressorPower} + \mathit{CondenserFanPower}}. \tag{1}$$

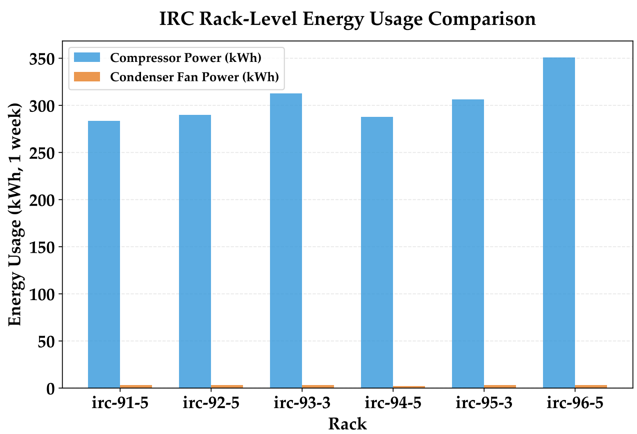


**Figure 5: Weekly compressor and condenser fan energy consumption for each in-row cooling (IRC) unit over a representative one-week window with complete telemetry. Compressor energy dominates total cooling-system input across all monitored zones.**

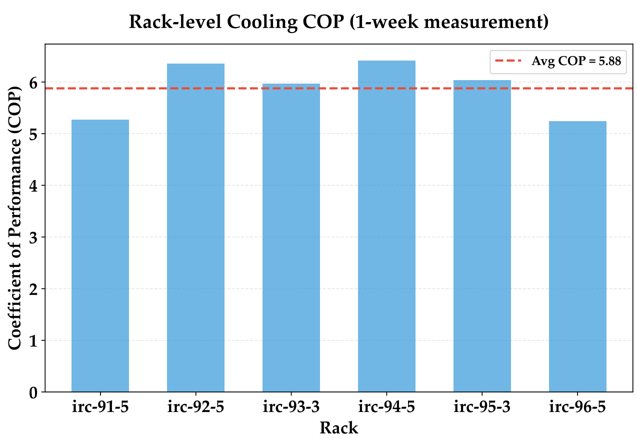


**Figure 6: Derived rack-level cooling coefficient of performance (COP) for each IRC unit over the same one-week window. The average COP across all units is 5.88, indicating stable and efficient cooling behavior.**

Figure 5 shows cooling-system energy usage across six IRC units during a one-week period selected to ensure continuous and complete telemetry coverage. Across all cooling zones, compressor power dominates total electrical input, while condenser fan power contributes only a small fraction, consistent with expected in-row cooling operation. Figure 6 shows the corresponding derived COP values, which remain stable across units with an average of 5.88 over the selected interval, indicating efficient and consistent cooling performance at the rack level.

## 4.3 Node Level Power Behavior

To understand how power is distributed within individual compute nodes, we analyze static power profiles under idle and peak operating conditions for representative CPU and GPU nodes.

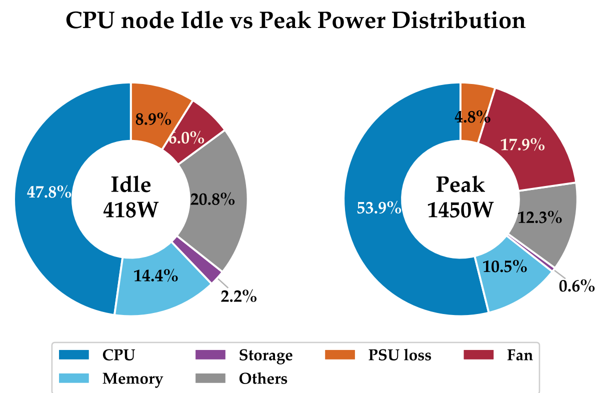


**Figure 7: Idle and peak power distribution for a representative CPU node, showing the contribution of major components including CPU, memory, fans, power supply losses, and other system components.**

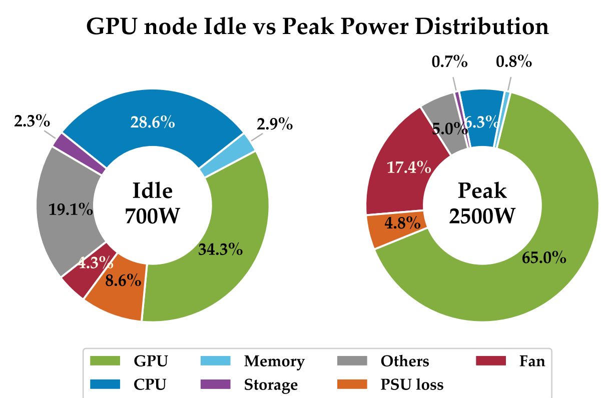


**Figure 8: Idle and peak power distribution for a representative GPU node, showing the contribution of major components including GPU, CPU, memory, fans, power supply losses, and other system components.**

Figure 7 shows the power composition of a representative CPU node under idle and peak conditions. At idle, power is dominated by baseline system components, while peak operation is primarily driven by increased CPU activity and fan power. As utilization rises, fan power becomes a significant secondary contributor, reflecting higher cooling demand under sustained compute load.

Figure 8 shows a markedly different profile for GPU nodes. Idle power is substantially higher due to the presence of accelerator hardware and associated cooling overhead. Under peak load, total system power increases sharply, with GPUs accounting for the majority of consumption. Fan power and power supply losses also grow noticeably, indicating that cooling and power delivery overheads form a larger fraction of total energy compared to CPU nodes.

These observations highlight that GPU nodes incur both higher idle baselines and larger non-compute energy overheads. Such characteristics have important implications for energy-aware scheduling, as consolidating workloads and minimizing idle time on accelerator nodes can yield disproportionate energy savings at the system level.

## 4.4 Application Level Energy Analysis

This subsection shifts the focus to a user perspective by examining the energy behavior of representative HPC applications. Energy is attributed using job-aligned out-of-band telemetry, but the analysis is organized around application workloads rather than scheduling constructs. We analyze five representative HPC applications executed on a single CPU node. Each application is run ten times, and node level telemetry is aggregated over the execution interval to obtain energy breakdowns across major system components.

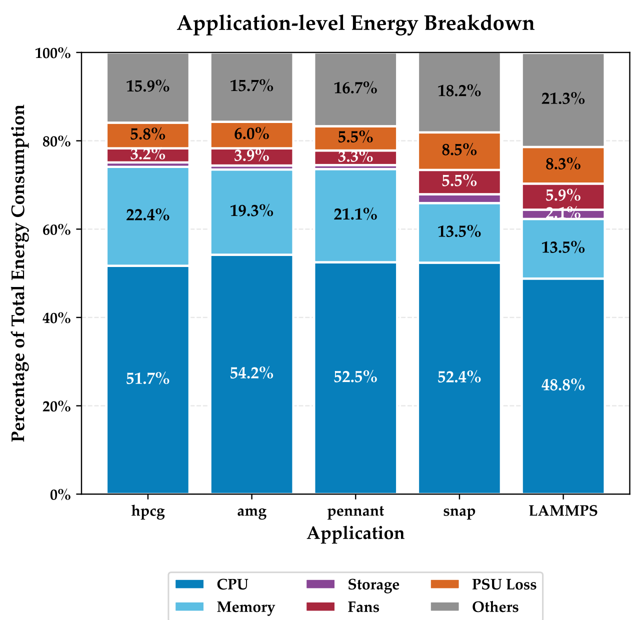


**Figure 9: Component-wise energy breakdown across representative CPU applications.**

Figure 9 shows that CPU energy dominates total consumption across all applications, accounting for roughly 49–54%. Memory energy forms a substantial secondary component (13–22%), with memory-intensive workloads exhibiting the highest shares. Fan energy remains relatively stable (3–6%), while power supply conversion losses contribute 5–9% of total energy. The remaining energy, grouped as "Others," reflects uncore activity and system overheads and accounts for 16–21%, with higher fractions observed in control-heavy workloads.

These results indicate that a significant portion of application energy is consumed outside the CPU cores themselves. Memory subsystems, uncore activity, cooling, and power delivery losses together represent a non-trivial share of total energy, underscoring the need for component-aware energy analysis beyond CPU-centric metrics.

# 5 Related Work

Energy-aware operation of large scale HPC systems has motivated extensive research on power monitoring infrastructures and power measurement methodologies. Cluster-scale monitoring frameworks such as LDMS [1] and DCDB/Wintermute [14] show scalable, high-frequency telemetry collection and aggregation, primarily targeting administrator-oriented monitoring and online analytics. MonSTer [13] further shows the practicality of correlating scheduler metadata with out-of-band telemetry obtained via Redfish, enabling low overhead job correlation using BMC sensors. Complementary studies compare power sensing interfaces, such as AC meters, IPMI/BMC sensors, and in-band counters such as RAPL and NVML, and report substantial variation in accuracy, granularity, and temporal stability across interfaces [7, 10]. Other work explores statistical or model based techniques to infer per process or per job energy consumption from coarse grained telemetry [4, 8, 12]. More recently, benchmark driven approaches have examined power behavior of GPU accelerated systems and AI workloads, including power-constrained throughput optimization [5] and energy-aware I/O and communication paths [9]. In contrast to these efforts, REPACSS focuses on *power observability* across multiple hardware layers by explicitly composing cooling, rack, and node out-of-band measurements into a unified energy model that supports facility level efficiency analysis (e.g., PUE), cross-layer validation, and scalable, long-horizon querying.

Recent studies show that making energy information visible to users is important for energy-aware application optimization. Shin et al. [18] show that exposing low-overhead energy telemetry to users enables practical analysis of performance–energy trade-offs on leadership-class systems. In addition, tools such as ClusterCockpit [6] focus on visualizing job-level performance and power data through web-based interfaces. Recent studies further analyze energy consumption of AI and LLM workloads, particularly inference, highlighting the dominance of inference energy cost and the sensitivity to software stacks [17] and configuration choices [16].

These efforts typically rely on in-job instrumentation, specialized runtimes, or benchmark-style execution. REPACSS complements this line of work by enabling *transparent, opt-in* job level energy reporting through native Slurm epilog integration. Energy accounting is performed post-run using persistent out-of-band telemetry, without modifying applications or introducing new submission interfaces. As a result, the framework simultaneously supports infrastructure level energy analysis and routine job level energy exposure, making it well suited for production-oriented HPC systems and shared research platforms.

# 6 Conclusion

This paper presented a power-centric observability framework for heterogeneous HPC systems and evaluated its design through deployment on the NSF REPACSS system. By composing out-of-band telemetry from in-row cooling units, rack level power distribution units, and node level iDRAC interfaces, the framework provides a unified pipeline from raw power sensing to facility, rack, node, and application aligned energy views, without requiring intrusive instrumentation or modifications to user applications.

Using this infrastructure, we conducted a set of multi-scope energy analyses that demonstrate the value of cross layer power observability. Facility level analysis shows stable and competitive PUE behavior derived directly from cooling and power-delivery telemetry. Rack level results highlight efficient and stable in-row cooling operation. Node level profiling reveals clear differences between CPU and GPU based platforms in idle power, peak consumption, and cooling overhead. Finally, application level studies illustrate how workload characteristics shape component-wise energy distribution, showing that non-CPU components such as DRAM, uncore subsystems, and power conversion losses account for a significant fraction of total energy.

In summary, these results underscore that meaningful energy understanding in modern HPC systems requires coordinated analysis across compute, cooling, and power-delivery subsystems. The proposed framework enables such analysis by combining persistent out-of-band telemetry with a power-centric energy model and a unified query interface that decouples energy analysis from raw sensor access. Lightweight Slurm integration further makes job-aligned energy information routinely accessible in production environments. This work demonstrates that practical, cross-scope energy observability can be achieved using existing management interfaces, and provides a foundation for future work on energy-aware scheduling, system optimization, and workload characterization in heterogeneous HPC systems.

## Acknowledgments

This research is supported by the National Science Foundation, Office of Advanced Cyberinfrastructure (Award Number 2404438; Program Manager: Robert Chadduck). ChatGPT (version 5.2) was used to assist with grammatical revision and language refinement of this manuscript.

## References

[1] Anthony Agelastos, Benjamin Allan, Jim Brandt, Paul Cassella, Jeremy Enos, Joshi Fullop, Ann Gentile, Steve Monk, Nichamon Naksinehaboon, Jeff Ogden, et al. 2014. The Lightweight Distributed Metric Service: A Scalable Infrastructure for Continuous Monitoring of Large Scale Computing Systems and Applications. In *SC14: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 154–165. doi:10.1109/SC.2014.18

[2] Ghazanfar Ali, Jon Hass, Alan Sill, Elham Hojati, Tommy Dang, and Yong Chen. 2022. Redfish-Nagios: A Scalable Out-of-Band Data Center Monitoring Framework Based on Redfish Telemetry Model. In *Fifth International Workshop on Systems and Network Telemetry and Analytics* (Minneapolis, MN, USA) *(SNTA '22)*. 3–11. doi:10.1145/3526064.3534108

[3] Lukas Alt, Anara Kozhokanova, Thomas Ilsche, Christian Terboven, and Matthias S. Mueller. 2024. An experimental setup to evaluate rapl energy counters for heterogeneous memory. In *Proceedings of the 15th ACM/SPEC International Conference on Performance Engineering*. 71–82. doi:10.1145/3629526.3645052

[4] Oscar Antepara, Zhengji Zhao, Brian Austin, Nan Ding, Leonid Oliker, Nicholas J. Wright, and Samuel Williams. 2025. Benchmark-driven Models for Energy Analysis and Attribution of GPU-Accelerated Supercomputing. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 888–904. doi:10.1145/3712285.3759815

[5] Nan Ding, Oscar Antepara, Zhengji Zhao, Brian Austin, Leonid Oliker, Nicholas J. Wright, and Samuel Williams. 2025. Maximizing Power-Constrained Supercomputing Throughput. In *ISC High Performance 2025 Research Paper Proceedings*. 1–13. doi:10.23919/ISC.2025.11017728

[6] Jan Eitzinger, Thomas Gruber, Ayesha Afzal, Thomas Zeiser, and Gerhard Wellein. 2019. Clustercockpit—a web application for job-specific performance monitoring. In *2019 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 1–7. doi:10.1109/CLUSTER.2019.8891017

[7] Daniel Hackenberg, Thomas Ilsche, Robert Schöne, Daniel Molka, Maik Schmidt, and Wolfgang E. Nagel. 2013. Power Measurement Techniques on Standard Compute Nodes: A Quantitative Comparison. In *2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 194–204. doi:10.1109/ISPASS.2013.6557170

[8] Hongyu Hè, Michal Friedman, and Theodoros Rekatsinas. 2024. Energat: Fine-grained energy attribution for multi-tenancy. *ACM SIGENERGY Energy Informatics Review* 4, 3 (2024), 18–25. doi:10.1145/3757892.3757900

[9] Md Hasibul Jamil, MD S. Q. Zulkar Nine, and Tevfik Kosar. 2025. EMLIO: Minimizing I/O Latency and Energy Consumption for Large-Scale AI Training. In *Proceedings of the SC '25 Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC Workshops '25)*. 2022–2031. doi:10.1145/3731599.3767566

[10] Mathilde Jay, Vladimir Ostapenco, Laurent Lefèvre, Denis Trystram, Anne-Cécile Orgerie, and Benjamin Fichel. 2023. An Experimental Comparison of Software-based Power Meters: Focus on CPU and GPU. In *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. IEEE, 106–118. doi:10.1109/CCGrid57682.2023.00020

[11] Ali Khosravi, Oscar R. Sandoval, Melika Sadat Taslimi, Tiia Sahrakorpi, Gessica Amorim, and Juan Jose Garcia Pabon. 2024. Review of energy efficiency and technological advancements in data center power systems. *Energy and Buildings* 323 (2024), 114834. doi:10.1016/j.enbuild.2024.114834

[12] Luis G. León-Vega, Niccolò Tosato, and Stefano Cozzini. 2025. A Comprehensive Analysis of Process Energy Consumption on Multi-Socket Systems with GPUs. In *High Performance Computing*. Springer, Springer Nature Switzerland, 52–67. doi:10.1007/978-3-031-80084-9_4

[13] Jie Li, Ghazanfar Ali, Ngan V. T. Nguyen, Jon R. Hass, Alan Sill, Tommy Dang, and Yong Chen. 2020. MonSTer: An Out-of-the-Box Monitoring Tool for High Performance Computing Systems. In *2020 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 119–129. doi:10.1109/CLUSTER49012.2020.00022

[14] Alessio Netti, Micha Müller, Carla Guillen, Michael Ott, Daniele Tafani, Gence Ozer, and Martin Schulz. 2020. DCDB Wintermute: Enabling Online and Holistic Operational Data Analytics on HPC Systems. In *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing*. 101–112. doi:10.1145/3369583.3392674

[15] Hafiz Adnan Niaz, Ravi Reddy Manumachu, and Alexey Lastovetsky. 2025. Accurate and Reliable Energy Measurement and Modelling of Data Transfer Between CPU and GPU in Parallel Applications on Heterogeneous Hybrid Platforms. *IEEE Trans. Comput.* 74, 3 (2025), 1011–1024. doi:10.1109/TC.2024.3504262

[16] Chenxu Niu, Wei Zhang, Jie Li, Yongjian Zhao, Tongyang Wang, Xi Wang, and Yong Chen. 2026. TokenPowerBench: Benchmarking the Power Consumption of LLM Inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 32582–32590. doi:10.1609/aaai.v40i38.40535

[17] Chenxu Niu, Wei Zhang, Yongjian Zhao, and Yong Chen. 2025. Energy efficient or exhaustive? benchmarking power consumption of llm inference engines. *ACM SIGENERGY Energy Informatics Review* 5, 2 (2025), 56–62. doi:10.1145/3757892.3757900

[18] Woong Shin, Karl W. Schulz, Arthur F. Lorenzon, Matthias Maiterth, Bruno Villasenor Alvarez, Jordà Polo, Aditya Kashi, Hao Lu, Nicholson Koukpaizan, Antigoni Georgiadou, et al. 2025. Bridging the Gap: User-Centric Energy Monitoring for Policy-Driven Application Optimization in HPC Data Centers. In *Proceedings of the SC '25 Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 2007–2016. doi:10.1145/3731599.3767564

[19] Zeyu Yang, Karel Adamek, and Wesley Armour. 2023. Part-time Power Measurements: nvidia-smi's Lack of Attention. *arXiv preprint arXiv:2312.02741* (2023). doi:10.48550/arXiv.2312.02741