Heterogeneous Energy Characterization of GPU-Powered Datalog Engines

Ahmedur Rahman Shovon [1] · Yihao Sun [2] · Zhiling Lan [1,3] · Swann Perarnau [1] · Thomas Gilray [4] · Kristopher Micinski [5] · Michael E. Papka [1,3] · Sidharth Kumar [3]

[1]Argonne National Laboratory · [2]Utah State University · [3]University of Illinois Chicago · [4]Washington State University · [5]Syracuse University

EESP Workshop · SC 2026

Argonne NATIONAL LABORATORY

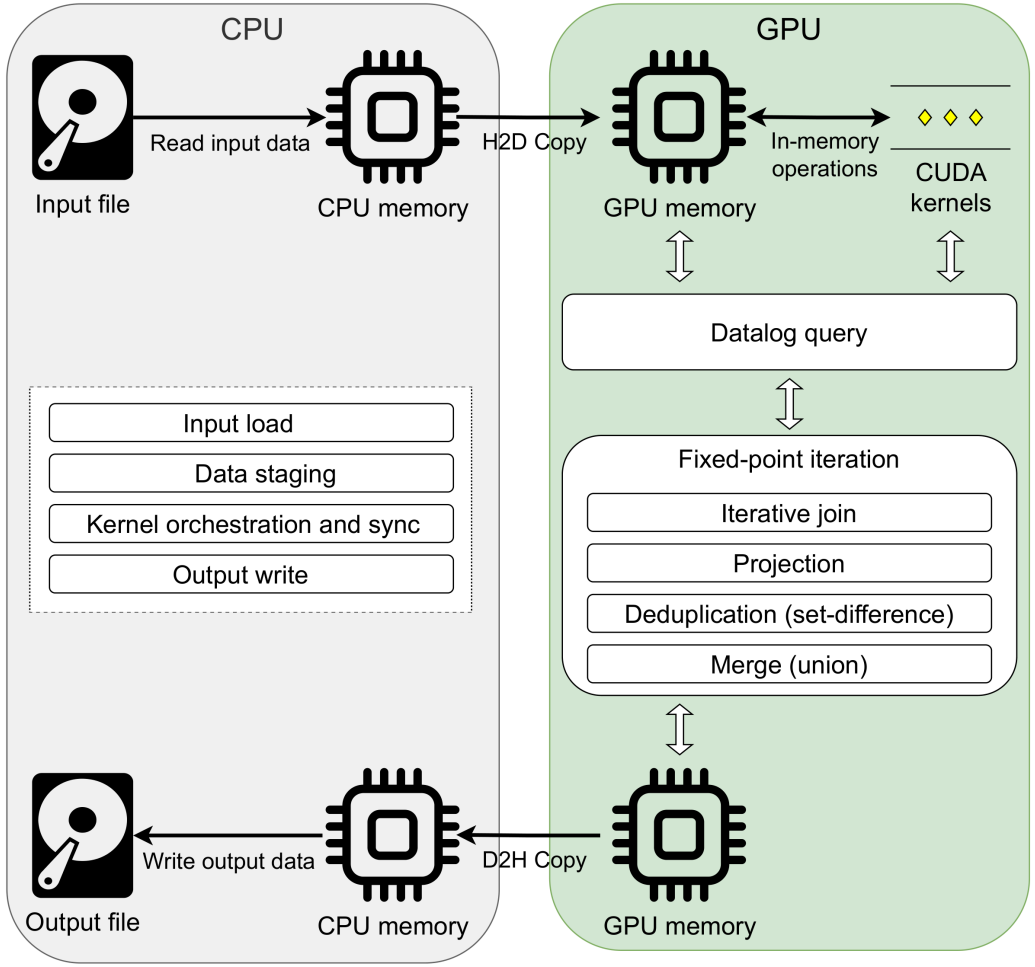
U.S. DEPARTMENT of ENERGY

GPU Datalog

- Datalog is a **logic programming language**: declarative recursive rules over relations
- Example: Reachability (transitive closure) in two rules:

```
T(x,y) :- G(x,y).
T(x,z) :- T(x,y), G(y,z).
```

- The engine applies the rules until no new fact appears, a **fixed point** of joins, deduplication, and unions
- Used in graph analytics, deductive databases[1], and program analysis[2]



1 *A. Shkapsky, M. Yang, M. Interlandi, H. Chiu, T. Condie, C. Zaniolo. Big Data Analytics with Datalog Queries on Spark. SIGMOD 2016*

2 *M. Bravenboer, Y. Smaragdakis. Strictly Declarative Specification of Sophisticated Points-to Analyses. OOPSLA 2009*

3 *A. R. Shovon, Y. Sun, K. Micinski, T. Gilray, S. Kumar. Multi-Node Multi-GPU Datalog. ICS 2025*

4 *Y. Sun, A. R. Shovon, T. Gilray, S. Kumar, K. Micinski. Optimizing Datalog for the GPU. ASPLOS 2025*

GPU Datalog engines (fast but energy-blind)

ENGINE	JOIN STRATEGY	DATA STRUCTURE
MNMGDatalog[1]	Radix hash join	GPU hash table (open addressing)
INLJoin	Index nested-loop join	GPU struct array
GPULog[2]	Sorted hash join	Hash-indexed sorted array (HISA)
BJoin	Batch join with memory pooling (RMM)	Hash-indexed sorted array (HISA)
cuDF[3]	DataFrame merge	Columnar GPU DataFrame

GPU engines[1,2] are up to **45x** faster than CPU based engine, but energy is never measured.

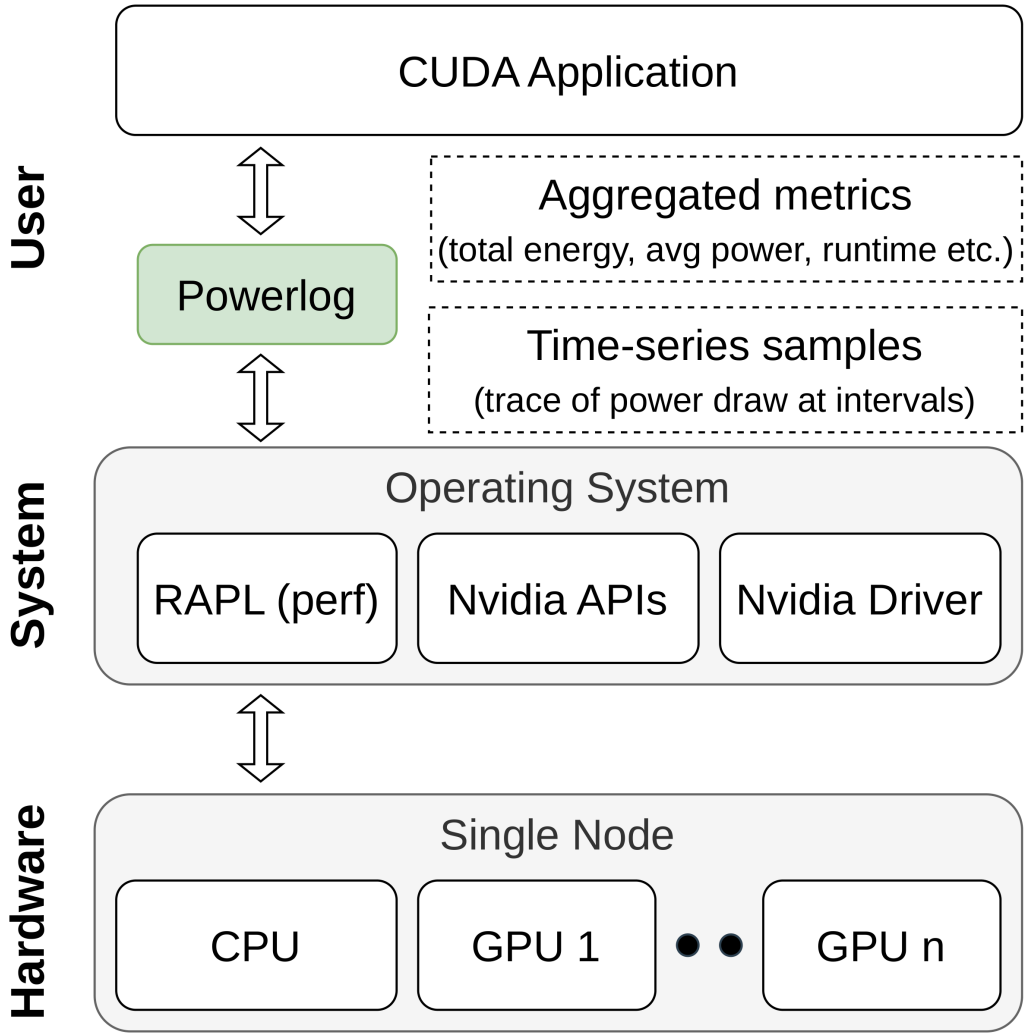
Recursive logic here is identical, performance and energy difference comes from the join strategy and data layout.

1 *A. R. Shovon, Y. Sun, K. Micinski, T. Gilray, S. Kumar. Multi-Node Multi-GPU Datalog. ICS 2025*

2 *Y. Sun, A. R. Shovon, T. Gilray, S. Kumar, K. Micinski. Optimizing Datalog for the GPU. ASPLOS 2025*

3 *A. R. Shovon, T. Gilray, K. Micinski, S. Kumar. Towards Iterative Relational Algebra on the GPU. USENIX ATC 2023*

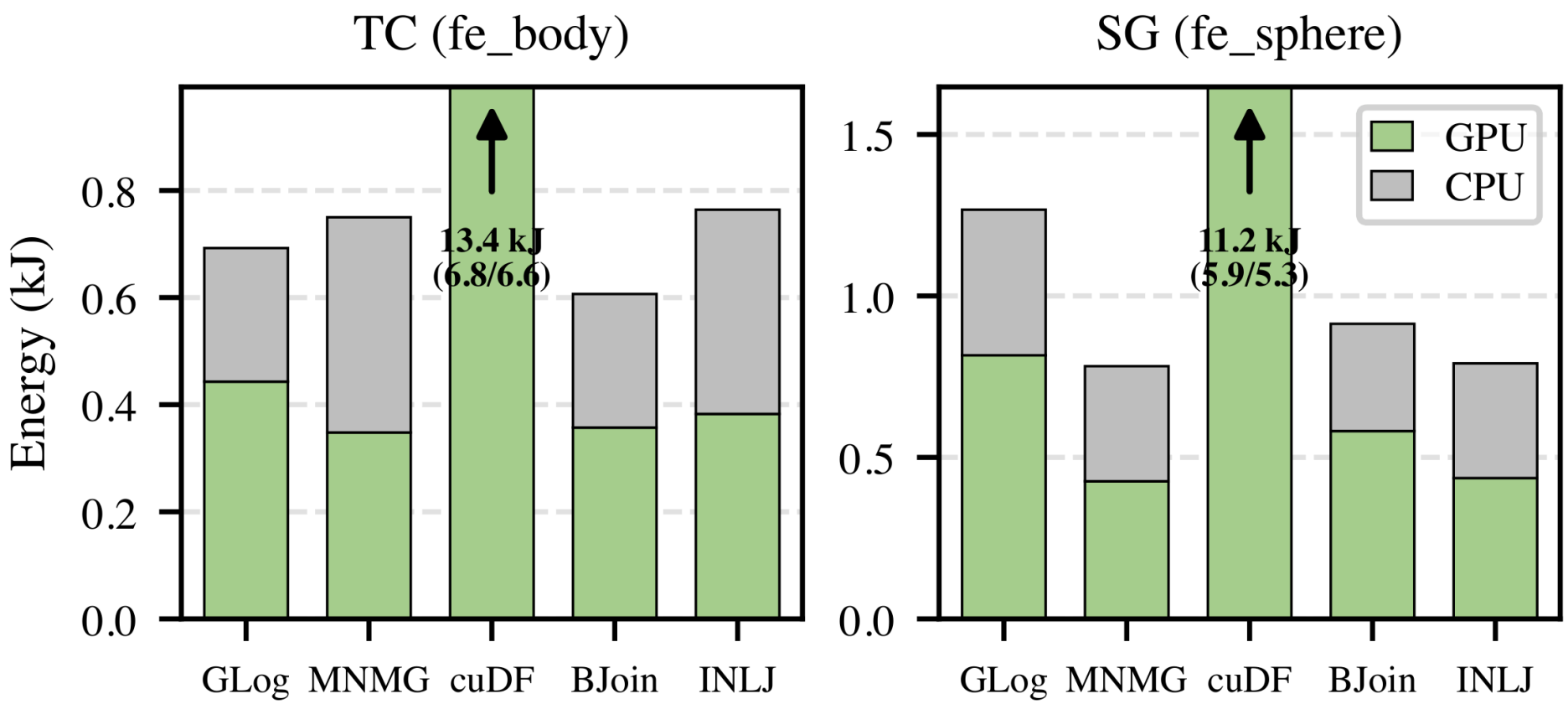
Heterogeneous power profiling (CPU to GPU) with Powerlog



Runs an unmodified program, samples GPU power via NVML and CPU package energy via RAPL

`pip install powerlog` Supports NVIDIA, AMD, and Intel GPUs. This paper reports NVIDIA results.

The host CPU is a first-class energy cost



Setup: AMD EPYC 7532 + NVIDIA A100 40GB (JLSE node, Argonne), 5 engines, TC and SG on SNAP[1] and SuiteSparse[2] graphs, mean of 10 runs

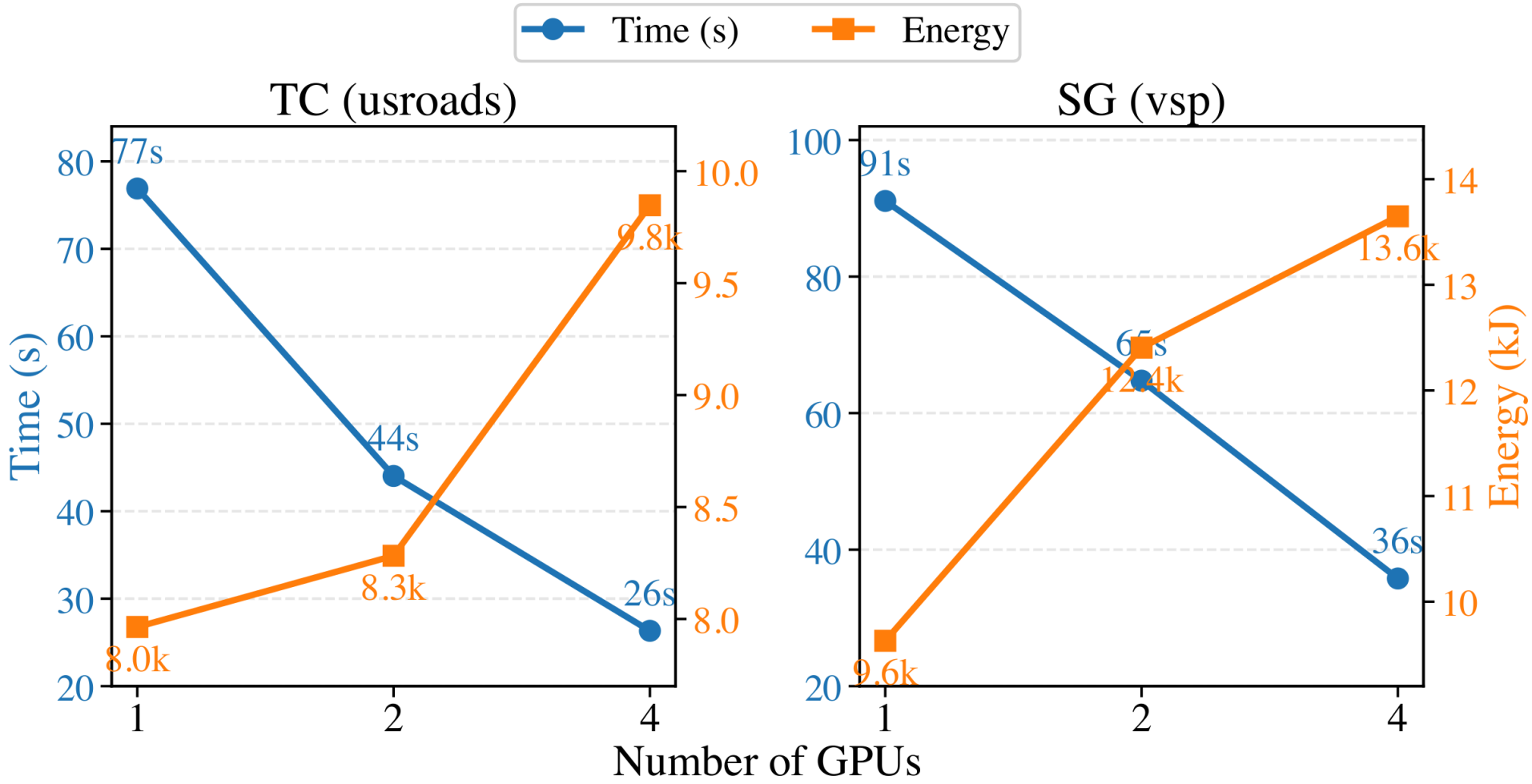
Host CPU is 32 to 55% of total energy, not a fixed overhead

GPU-only accounting understates energy by up to 2x

Engine choice alone shifts total energy up to 3x

[1] *J. Leskovec, A. Krevl. SNAP Datasets: Stanford Large Network Dataset Collection. 2014* [2] *T. A. Davis, Y. Hu. The SuiteSparse Matrix Collection. ACM TOMS 2011*

Scaling GPUs is faster but not greener



Setup: Polaris node (Argonne) with 4x NVIDIA A100, MNMGDatalog, GPU energy only

1 to 4 GPUs: runtime drops 2.5 to 2.9x, but GPU energy rises (TC +24%, SG +42%)

Single-GPU minimizes energy, multi-GPU minimizes EDP: fastest is not greenest

Summary

- Measure the **whole node**: the host CPU is **32 to 55%** of total energy
- **Engine design decides energy**: join strategy and data layout shift it up to **3x**
- Scaling 1 to 4 GPUs cuts runtime **2.9x** but raises energy **24 to 42%**
- First heterogeneous CPU and GPU characterization with multi vendor support
- Next: move from profiling to **power control**, and characterize fused-kernel GPU Datalog engine[1]

[1] *A. R. Shovon, R. Pereira, Y. Sun, F. Saif, S. Kumar, S. Perarnau. Accelerating GPU Datalog with Kernel Fusion. IA3 2026*

Acknowledgements

This work is a collaboration among **Argonne National Laboratory**, the **University of Illinois Chicago**, **Utah State University**, **Washington State University**, and **Syracuse University**.

We gratefully acknowledge the computing resources provided and operated by the **Joint Laboratory for System Evaluation (JLSE)** at Argonne National Laboratory. This research also used resources of the **Argonne Leadership Computing Facility**.

The facilities and contributions of Argonne National Laboratory are supported by the Office of Science of the U.S. Department of Energy under contract **DE-AC02-06CH11357**, and in part by the U.S. Department of Energy under Contract DE-SC0024271.

Supported in part by the U.S. National Science Foundation under grants **OAC 2402901, CCF 2515009, CCF 2316157**, and **CCF 2401274**.

Thank you

ashovon@anl.gov

Code
github.com/arsho/powerlog

Package
pypi.org/project/powerlog

Experimental setup

Hardware

- Single-GPU: JLSE node, AMD EPYC 7532 CPU, NVIDIA A100 40GB
- Multi-GPU: Polaris node, four NVIDIA A100 per node
- CPU energy from RAPL (perf), GPU energy from NVML

Software and workload

- CUDA 12.9.1, nvhpc 25.11 (HPCX MPI), gcc 12.2.0
- Five engines, two queries: TC and SG
- Datasets from SNAP[1] and SuiteSparse[2]

All energy and runtime values are the mean of **ten runs** after a warm-up round.

[1] *J. Leskovec, A. Krevl. SNAP Datasets: Stanford Large Network Dataset Collection. 2014* [2] *T. A. Davis, Y. Hu. The SuiteSparse Matrix Collection. ACM TOMS 2011*

Per-engine efficiency: energy and instructions per joule

DATASET (EDGES)	GLOG	MNMG	CUDF	BJOIN	INLJ	GLOG	MNMG	CUDF	BJOIN	INLJ
	GPU/CPU energy (J), lower is better					GPU instr per joule (M/J), higher is better				
Transitive Closure (TC)										
fe_body (163k)	**443/249**	**348/402**	**6814/6624**	**357/249**	**383/381**	**413**	**274**	**185**	**488**	**252**
usroads (165k)	**5885/2754**	**5091/5804**	**OOM**	**3821/1837**	**5052/5819**	**329**	**233**	**OOM**	**497**	**236**
Same Generation (SG)										
loc-brightkite (214k)	**1147/555**	**337/183**	**1303/1324**	**897/487**	**340/182**	**331**	**499**	**255**	**413**	**489**
fe_sphere (49k)	**816/451**	**426/356**	**5908/5281**	**582/332**	**436/355**	**377**	**333**	**189**	**515**	**333**

Engine choice shifts total energy up to 3x · cuDF runs ~10x more instructions (more work, not just waste) · no engine is universally best: BJoin wins TC, MNMG wins SG

Measured with Powerlog on NVIDIA A100 (JLSE)