Scalable Multi-Node Multi-GPU Datalog Engine with Energy-Aware Profiling

Ahmedur Rahman Shovon
ashovon@anl.gov
Argonne National Laboratory
Lemont, Illinois, USA

Sidharth Kumar
sidharth@uic.edu
University of Illinois, Chicago
Chicago, Illinois, USA

1 Introduction

Datalog is a rule-based symbolic AI language [1] well suited for complex recursive queries across data-intensive applications. Its declarative rules have enabled advances in large-scale data analytics [2–5], program analysis and verification [6–9], and recently neuro-symbolic reasoning [10]. In graph mining, canonical applications include Transitive Closure (TC), Same Generation (SG), and Connected Components (CC), all expressible in a few lines of rules. Engines compile these rules into relational algebra operators (joins, projections, set differences) executed iteratively until reaching a fixed point [11, 12].

Established engines such as Soufflé [13] and distributed frameworks like SLOG [14] target multi-core CPUs. Recently, GPUs have attracted attention due to massive parallelism and high memory bandwidth, which align with Datalog's iterative joins, deduplication, and set operations. GPU-accelerated engines adopt diverse strategies: cuDF-based systems [15] leverage Python data libraries, while GPUJoin[16] implements custom CUDA kernels. Engines such as GPULog[17] refine GPU data structures for state-of-the-art performance. However, none support distributed multi-node, multi-GPU (MNMG) platforms central to modern HPC systems. Moreover, nearly all prior work emphasizes throughput, leaving the *energy efficiency* of GPU-powered Datalog largely unexplored.

2 Implementation

We present mnmgDatalog [18], the first Datalog engine for distributed MNMG environments. It addresses scalability challenges of recursive workloads on heterogeneous clusters. mnmgDatalog exploits CUDA for intra-node parallelism across GPUs and MPI for inter-node coordination. Three innovations enable efficient fixed-point evaluation:

1. GPU-parallel joins: essential for recursive rules over large intermediate relations.
2. Scalable recursive aggregation: handling workloads where tuple counts fluctuate across iterations.
3. Iterative all-to-all communication: reducing synchronization overhead via GPU-aware communication.

Equally critical is energy efficiency. While runtime has long been the main metric, power consumption increasingly limits exascale systems. GPUs provide high speedups but draw significant power, raising efficiency concerns. Despite widespread GPU adoption, no prior work has examined Datalog's energy behavior. To fill this gap, we developed Powerlog, the first GPU energy profiler for Datalog. It is a lightweight user-space wrapper that runs a target binary while sampling GPU power at fixed intervals. It aggregates readings across devices and reports runtime, average power, and energy consumed. Unlike FLOPS/watt used in benchmarks such as Green500 [19], relational Datalog computation requires a domain-specific measure. We therefore propose *tuples per joule*, quantifying the number of result tuples produced per unit energy. This metric captures the cost of useful work in recursive query processing and enables fair comparison across designs.

3 Evaluation

We evaluated mnmgDatalog on Argonne's Polaris supercomputer, featuring AMD EPYC CPUs and NVIDIA A100 GPUs interconnected by NVLink. Benchmarks included TC, SG, and CC, scaled from one GPU to 32 GPUs across nodes. We compared against single-GPU, shared-memory, and distributed multi-core engines. For power profiling, we scale execution from one to four GPUs within a single node. Datasets are drawn from the Stanford Large Network Dataset Collection [20] and the SuiteSparse Matrix Collection [21].

Results show mnmgDatalog gains:

- Up to 7× speedup over single-GPU engines.
- Up to 33× speedup over multicore engine (Soufflé).
- Up to 32× speedup over distributed engine (SLOG).

Energy analysis with Powerlog reveals clear tradeoffs. On a single node, scaling mnmgDatalog from 1 to 4 GPUs reduces runtime almost linearly, yet total energy consumption rises. This corresponds to a 23% increase for TC and a 42% increase for SG. Our tuples-per-joule evaluation shows that energy efficiency consistently declines with additional GPUs, with drops of nearly 30%. These results indicate that throughput gains from parallelism are offset by higher per-node power draw and communication overheads. Notably, the relative decrease is similar for both TC and SG, suggesting a hardware-driven effect rather than workload-specific behavior. Thus, while multi-GPU runs remain advantageous for latency-sensitive tasks, the highest energy efficiency is achieved in single-GPU execution.

By combining performance and energy perspectives, our study highlights the tradeoff between *time-to-solution* and *energy-to-solution*. Future exascale workloads will demand systems that deliver both speed and efficiency. With MNMGDATALOG and POWERLOG, we establish a framework for making such informed choices.

4 Conclusion

This poster contributes along two dimensions. First, it introduces MNMGDATALOG, the first MNMG Datalog engine, which highest performant Datalog engine to date. Second, it presents POWERLOG, the first GPU energy profiler for Datalog, enabling benchmarking with the tuples/joule metric. These contributions advance declarative analytics toward energy-aware execution on heterogeneous exascale platforms. Future work on MNMGDATALOG will focus on adaptive load balancing, portability across HIP and OneAPI, and applications in emerging domains such as neurosymbolic programming and topological data analysis. In parallel, we will extend POWERLOG to multi-node environments to enable whole-system energy evaluation of distributed GPU clusters.

References

[1] Ziyang Li, Jiani Huang, Jason Liu, Felix Zhu, Eric Zhao, William Dodds, Neelay Velingker, Rajeev Alur, and Mayur Naik. Relational programming with foundation models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024.

[2] Alexander Shkapsky, Mohan Yang, Matteo Interlandi, Hsuan Chiu, Tyson Condie, and Carlo Zaniolo. Big data analytics with datalog queries on spark. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, page 1135–1149, New York, NY, USA, 2016. Association for Computing Machinery.

[3] Jiaqi Gu, Yugo H Watanabe, William A Mazza, Alexander Shkapsky, Mohan Yang, Ling Ding, and Carlo Zaniolo. Rasql: Greater power and performance for big data analytics with recursive-aggregate-sql on spark. In *Proceedings of the 2019 International Conference on Management of Data*, pages 467–484, 2019.

[4] Jiwon Seo, Stephen Guo, and Monica S Lam. Socialite: Datalog extensions for efficient social network analysis. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 278–289. IEEE, 2013.

[5] Evgeny Skvortsov, Yilin Xia, and Bertram Ludäscher. Logica: Declarative data science for mere mortals. In *EDBT*, pages 842–845, 2024.

[6] Martin Bravenboer and Yannis Smaragdakis. Strictly declarative specification of sophisticated points-to analyses. In *Proceedings of the 24th ACM SIGPLAN conference on Object oriented programming systems languages and applications*, pages 243–262, 2009.

[7] Antonio Flores-Montoya and Eric Schulte. Datalog disassembly. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1075–1092, 2020.

[8] George Balatsouras and Yannis Smaragdakis. Structure-sensitive points-to analysis for c and c++. In *Static Analysis: 23rd International Symposium, SAS 2016, Edinburgh, UK, September 8-10, 2016, Proceedings 23*, pages 84–104. Springer, 2016.

[9] Zhiwei Fan, Jianqiao Zhu, Zuyu Zhang, Aws Albarghouthi, Paraschos Koutris, and Jignesh M. Patel. Scaling-up in-memory datalog processing: Observations and techniques. *Proc. VLDB Endow.*, 12(6):695–708, Feb 2019.

[10] Adithya Murali, Atharva Sehgal, Paul Krogmeier, and P Madhusudan. Composing neural learning and symbolic reasoning with an application to visual discrimination. *arXiv preprint arXiv:1907.05878*, 2019.

[11] Sidharth Kumar and Thomas Gilray. Distributed relational algebra at scale. In *International Conference on High Performance Computing, Data, and Analytics (HiPC). IEEE*, volume 1, 2019.

[12] Sidharth Kumar and Thomas Gilray. Load-balancing parallel relational algebra. In *High Performance Computing: 35th International Conference, ISC High Performance 2020, Frankfurt/Main, Germany, June 22–25, 2020, Proceedings 35*, pages 288–308. Springer, 2020.

[13] Herbert Jordan, Bernhard Scholz, and Pavle Subotić. Soufflé: On synthesis of program analyzers. In *Computer Aided Verification: 28th International Conference, CAV 2016, Toronto, ON, Canada, July 17-23, 2016, Proceedings, Part II 28*, pages 422–430. Springer, 2016.

[14] Thomas Gilray, Arash Sahebolamri, Yihao Sun, Sowmith Kunapaneni, Sidharth Kumar, and Kristopher Micinski. Datalog with first-class facts. *Proceedings of the VLDB Endowment*, 18(3):651–665, 2024.

[15] Ahmedur Rahman Shovon, Landon Richard Dyken, Oded Green, Thomas Gilray, and Sidharth Kumar. Accelerating datalog applications with cudf. In *2022 IEEE/ACM Workshop on Irregular Applications: Architectures and Algorithms (IA3)*, pages 41–45. IEEE, 2022.

[16] Ahmedur Rahman Shovon, Thomas Gilray, Kristopher Micinski, and Sidharth Kumar. Towards iterative relational algebra on the {GPU}. In *2023 USENIX Annual Technical Conference (USENIX ATC 23)*, pages 1009–1016, 2023.

[17] Yihao Sun, Ahmedur Rahman Shovon, Thomas Gilray, Sidharth Kumar, and Kristopher Micinski. Optimizing datalog for the gpu. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, pages 762–776, 2025.

[18] Ahmedur Rahman Shovon, Yihao Sun, Kristopher Micinski, Thomas Gilray, and Sidharth Kumar. Multi-node multi-gpu datalog. In *Proceedings of the 2025 International Conference on Supercomputing (ICS '25)*, page 15, New York, NY, USA, 2025. ACM.

[19] Green500 list, june 2025. https://top500.org/lists/green500/2025/06/, June 2025. Copyright 1993–2025 TOP500.org. Accessed: July 19, 2025.

[20] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[21] Timothy A. Davis and Yifan Hu. The university of florida sparse matrix collection. *ACM Trans. Math. Softw.*, 38(1), dec 2011.