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

Ahmedur Rahman Shovon (ashovon@anl.gov)
Sidharth Kumar (sidharth@uic.edu)

MNMGDatalog | Powerlog

Introduction

Declarative programming focuses on "WHAT" not on "HOW"

Users

UserID	UserName	UserEmail	Country
101	Alice	alice@example.com	USA
102	Bob	bob@example.com	USA
103	Eve	eve@example.com	Canada

WHAT **SELECT** UserID **FROM** Users **WHERE** Country='USA'; HOW

Advanced approach: Logic programming (Datalog)

Datalog rules to compute Transitive Closure (TC) of a relation

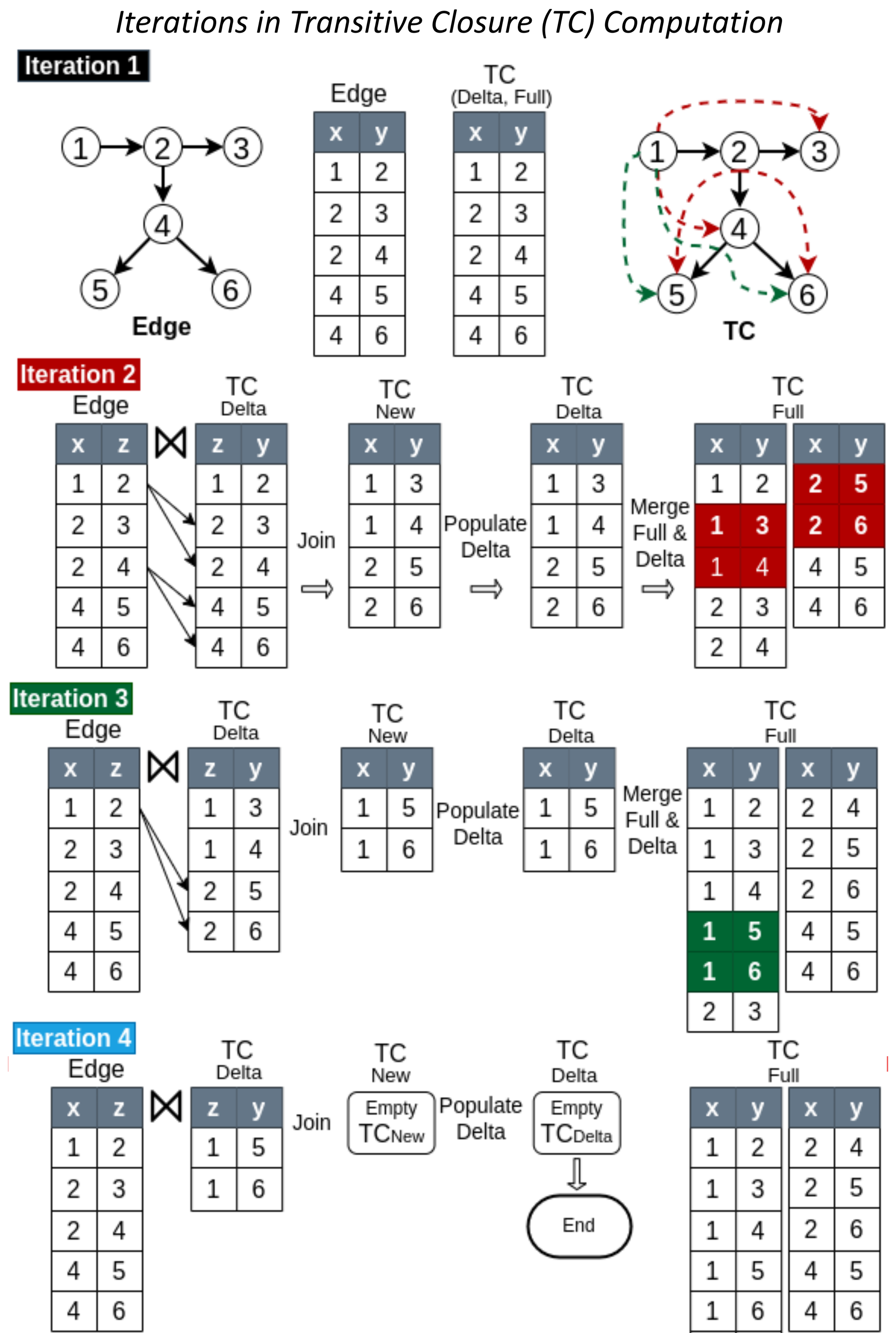
```
TC(x, y) :- Edge(x, y).
TC(x, z) :- TC(x, y), Edge(y, z).
```

Operationalized as a **fixed-point iteration** using F_G

$$F_G(T) \triangleq G \cup \Pi_{1,2}(\rho_{0/1}(T) \bowtie_1 G)$$

Datalog rules compiled down to **iterative relational algebra**

Iterations in Transitive Closure (TC) Computation



Datalog Engine Categories

- Multi-threaded: Soufflé, LogicBlox
- Distributed (Apache Spark): RDFox, Radlog
- Multi-node Multi-threaded: SLOG, PRAM
- Single-GPU: GPUJoin, GPULog
- Multi-node Multi-GPUs (MNMG): ?

Requirements for MNMG Datalog Engine

1. Workload partitioning
2. Data representation
3. Efficient communication
4. Tuple materialization

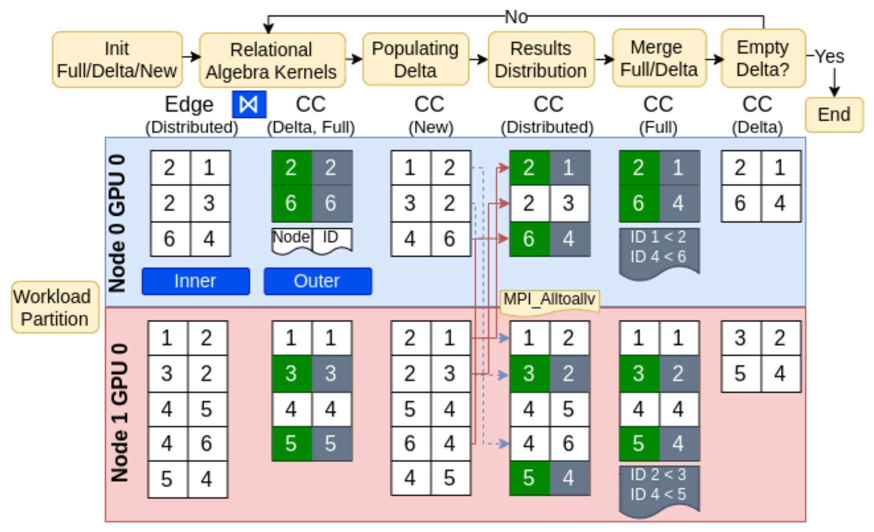
MNMGDatalog is the first MNMG Datalog engine

First multi-node multi-GPU Datalog engine
Single-GPU: Up to 7× speedup over GPULog
Multi-threaded: Up to 33× over Soufflé
Distributed: Up to 31.9× speedup over SLOG

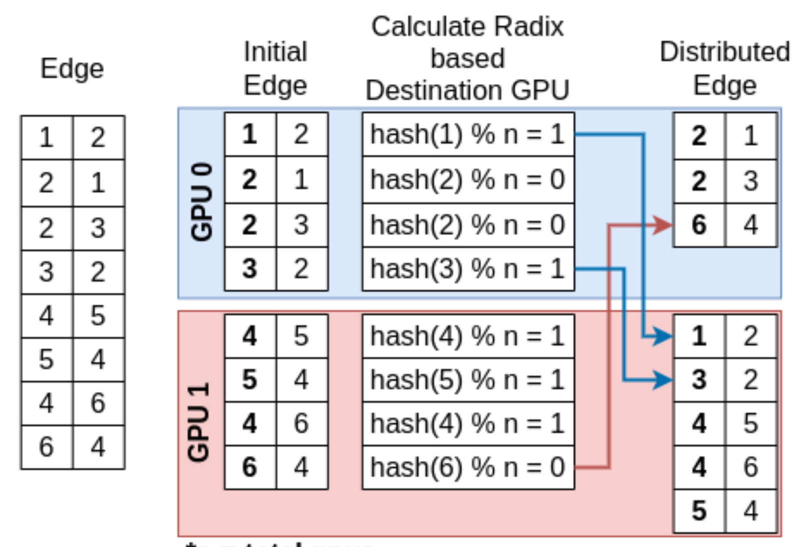
MNMGDatalog Implementation

MNMGDatalog uses radix-hash partitioning and non-uniform all-to-all communication with GPU-aware hash tables for efficient tuple materialization

Materialization in fixed point iteration



Radix-hash-based data partitioning



*n = total gpus

Performance Experiments

We evaluate **MNMGDatalog** against state-of-the-art single-GPU, shared-memory, and distributed multi-node Datalog engines up to 32 NVIDIA A100 GPUs

Experiment platform, application, and datasets

- **Polaris** supercomputer from **Argonne National Lab**
- **HW**: AMD EPYC 7543P CPU, NVIDIA A100 GPUs with NVLink
- **Apps**: Transitive Closure, Same Generation, Connected Component
- **Datasets**: Stanford large network, SuiteSparse, Road network

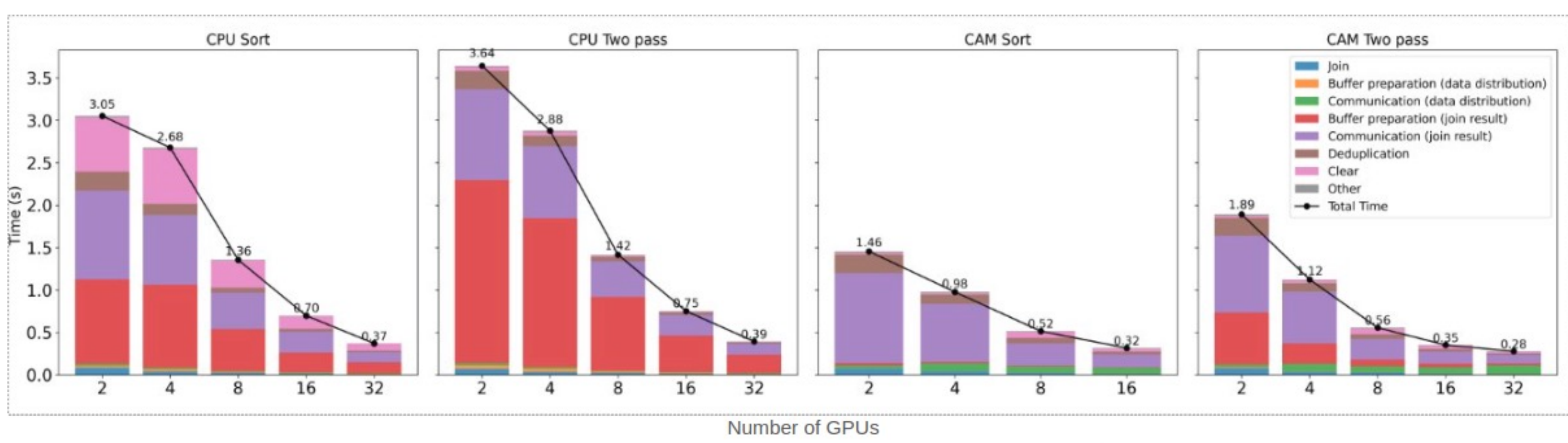
Single-GPU evaluation for Transitive Closure (TC)

Dataset name	TC edges	MNMGDatalog	GPULog	Soufflé	GPUJoin
		Time (s)			
com-dblp	1.91B	**13.58**	26.95	232.99	OOM
fe_ocean	1.67B	**66.34**	72.74	292.15	100.30
usroads	871M	**75.07**	78.08	222.76	364.55
vsp_finan	910M	**81.14**	82.75	239.33	125.94
Gnutella31	884M	**4.75**	7.64	96.82	OOM

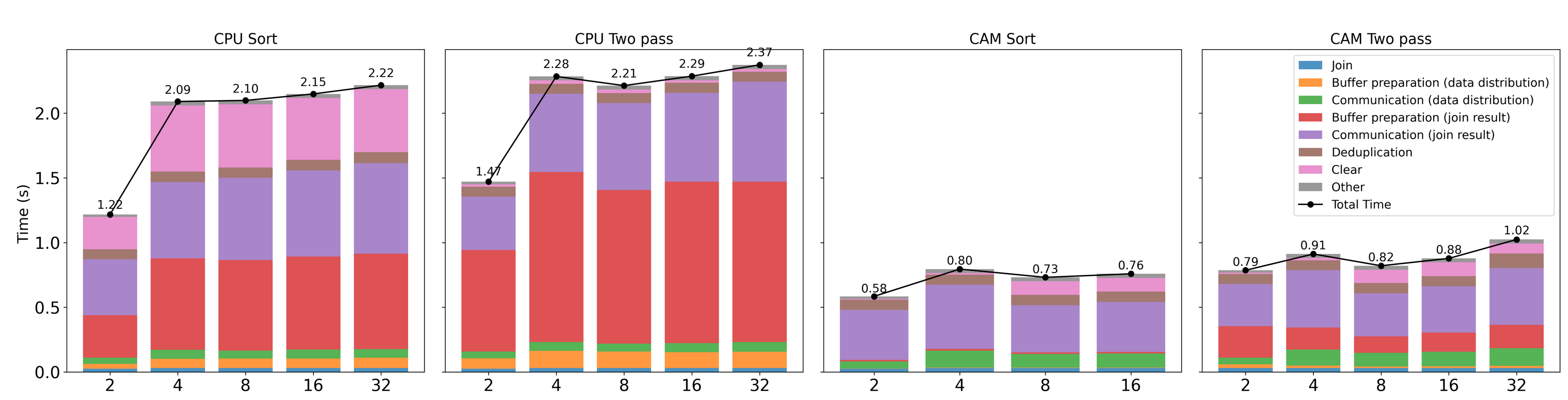
Single-GPU evaluation for Same Generation (SG)

Dataset name	SG size	MNMGDatalog	GPULog	Soufflé	cuDF
		Time (s)			
fe_body	408M	**9.08**	18.41	74.26	OOM
loc-Brightkite	92.3M	**1.66**	11.67	48.18	OOM
fe_sphere	205M	**3.55**	7.88	48.12	OOM
CA-HepTH	74M	**0.60**	4.79	20.12	21.24

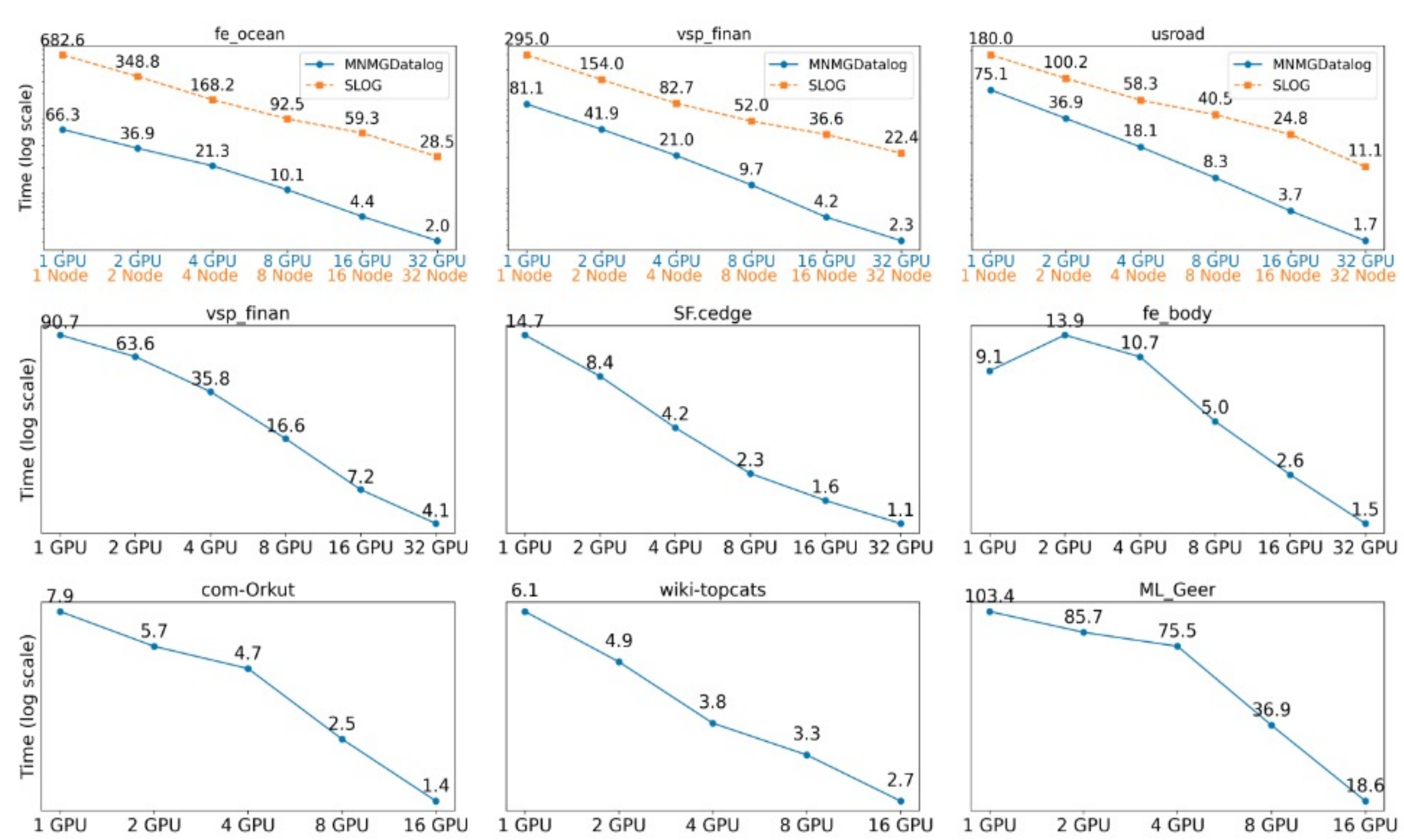
Strong scaling for iterative join (total 10M tuples)



Weak scaling for iterative join (10M tuples/GPU)



Multi-node evaluation for TC, SG, and CC



Powerlog for Energy Profiling

Powerlog: the first GPU energy profiler for Datalog engines, enabling energy-efficiency evaluation of SOTA GPU-powered engines on Polaris

Power consumption measurement

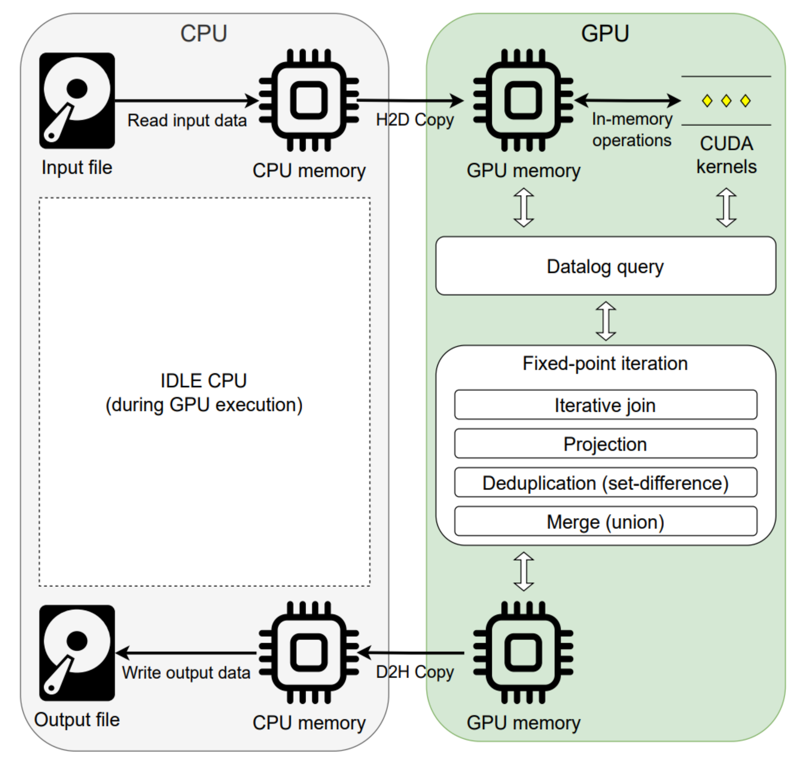
$$E = \int_0^T P(t)\,dt \approx \sum_{i=1}^{N} P_i \cdot \Delta t_i$$

where, E = total energy, N = # of samples, P_i = power draw at i, Δt_i = elapsed time between i,i+1

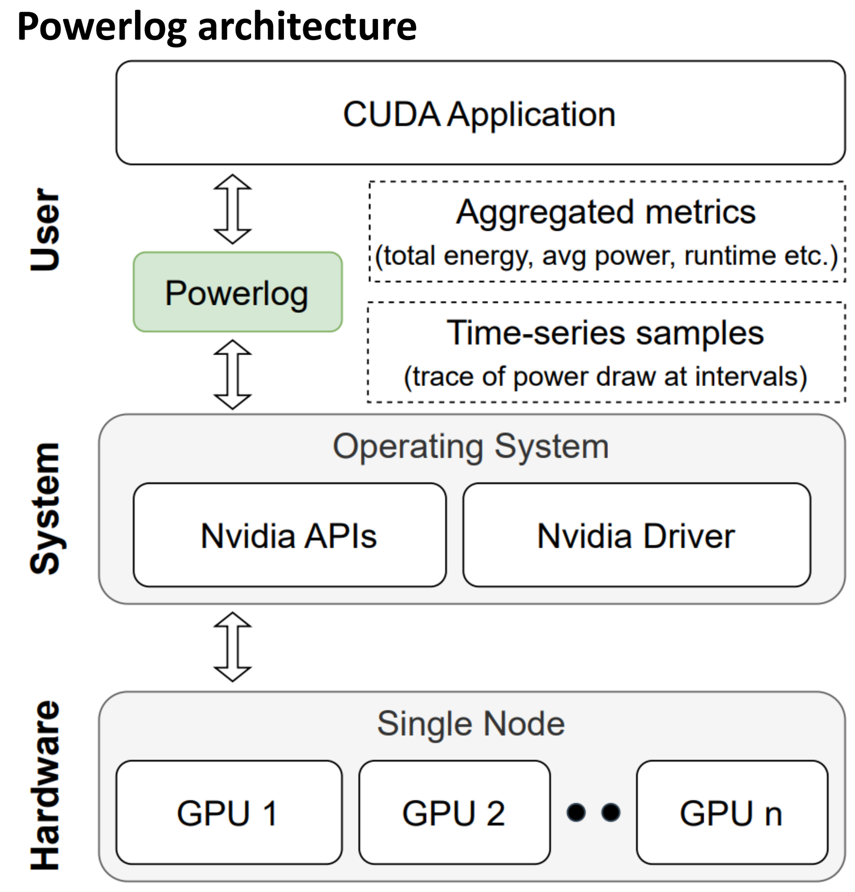
Energy efficiency metric: Tuples/Joule

$$\text{Tuples per Joule} = \frac{\text{Total number of tuples produced}}{\text{Total energy consumed (Joules)}}$$

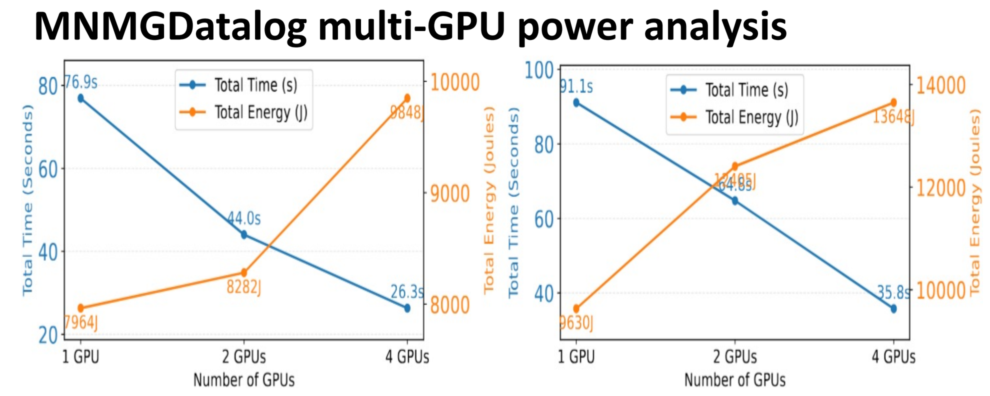
Workflow for energy profiling Datalog engines



Powerlog architecture



MNMGDatalog multi-GPU power analysis



Energy efficiency evaluation using Tuples/Joule

Application	Dataset	1 GPU	2 GPUs	4 GPUs
Transitive Closure	usroads	**109409**	105207	88483
Same Generation	vsp	**89802**	69710	63361

Conclusion

- First ever Datalog engine designed for multi-node multi-GPU HPC systems outperforming state-of-the-art engines
- Introduces novel GPU-Aware communication for scalable recursive query evaluation
- Supports recursive aggregation for Datalog rules using high-throughput GPU kernels
- Power profiling of GPU-based Datalog engines

Engine evolution

- Accelerating Datalog applications with cuDF — IA³ 2022
- Towards iterative relational algebra on the (GPU) — USENIX ATC '23
- Optimizing Datalog for the GPU — ASPLOS 2025
- Multi-node multi-GPU Datalog — ICS

Acknowledgement

NSF · Argonne National Laboratory · The HARP Lab

Future plan

- Spatial and temporal load balancing
- GPU-Aware HIP and OneAPI implementations
- Application to Neurosymbolic programming
- Extend **Powerlog** to multi-node environment