Optimizing Parallel Discrete Event Simulations with Profile-Guided Partitioning

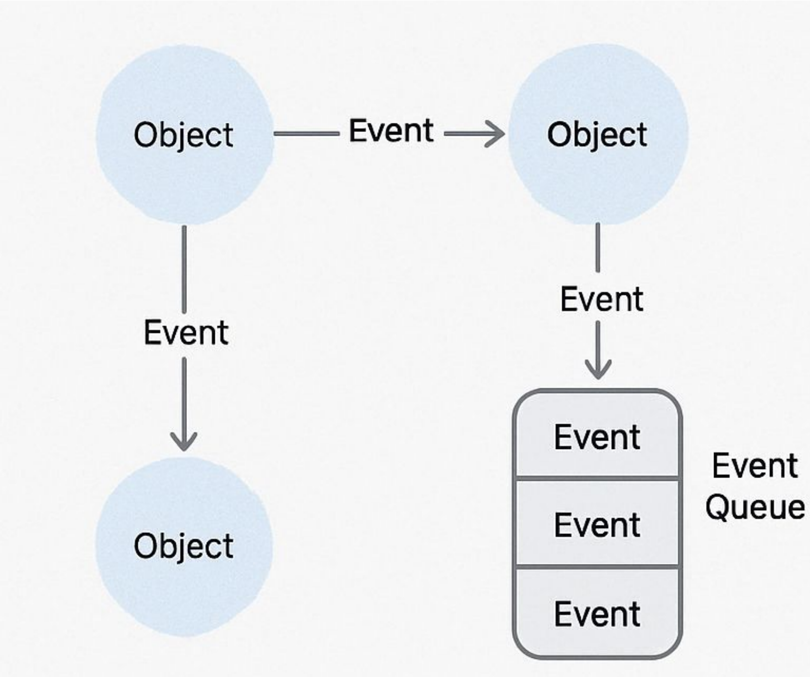
Maram Sunil Reddy
IXPUG 2025 | smaram1370@gmail.com

Agenda

- Introduction to Discrete Event Simulation (DES)
- Parallel DES and communication challenges
- Profile-Guided Partitioning: The method
- Benchmarks and experimental setup
- Performance analysis
- Conclusions and future work

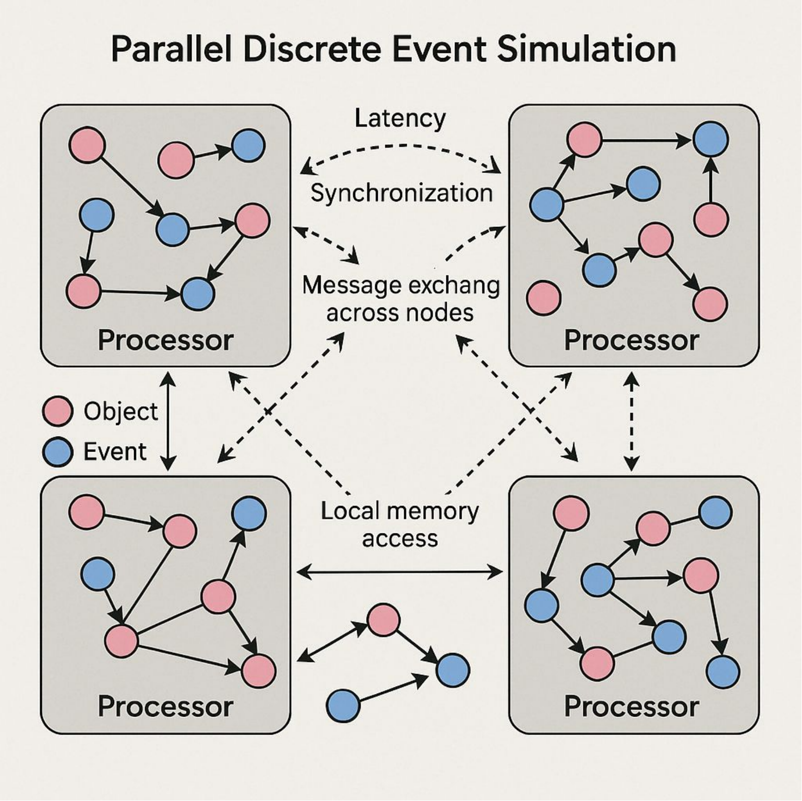
What is Discrete Event Simulation, or DES?

- Simulation technique where system state changes only at discrete events
- Efficient for modeling digital systems, networks, manufacturing, etc.



Why Parallelize DES?

- Large models → long runtimes
- Parallel DES (PDES) enables scalability
- But... communication becomes a bottleneck



The Communication Challenge

- Inter-process communication is expensive
- Especially over networked nodes
- Bottlenecks reduce expected speedups

Problem Statement

- How to reduce network traffic in PDES?
- How to maintain balanced computational loads?

Related Work – Profiling Approaches

“Several researchers have explored **profiling** as a way to guide simulation optimization. Notably:

- **Bahulkar et al.** introduced dynamic profiling for partition adaptation during simulation.
- **Guo and Hu** used fire spread models to create profile-based spatial partitioning — but their approach was tailored to a single application domain.
- **Peschlow et al.** proposed flexible dynamic partitioning but required runtime system integration and feedback loops.

Our work is different in key ways:

1. **offline profiling**
2. **general-purpose models**
3. **well-established, fast partitioner (METIS)** — no custom tools needed.

Our Solution

Profile-Guided Partitioning

- Uses communication data from a sequential simulation
- Clusters frequently interacting simulation objects
- Minimizes cross-node communication

Concept Overview

- Simulation objects = nodes
- Events exchanged = weighted edges
- Goal: Partition graph to reduce edge cuts

Profiling Phase

- Run model sequentially
- Record message count between object pairs
- Build communication graph

Heatmap Example

- ISCAS'89 s9234 model
- High-frequency communication visualized
- Dense clusters of interaction identified

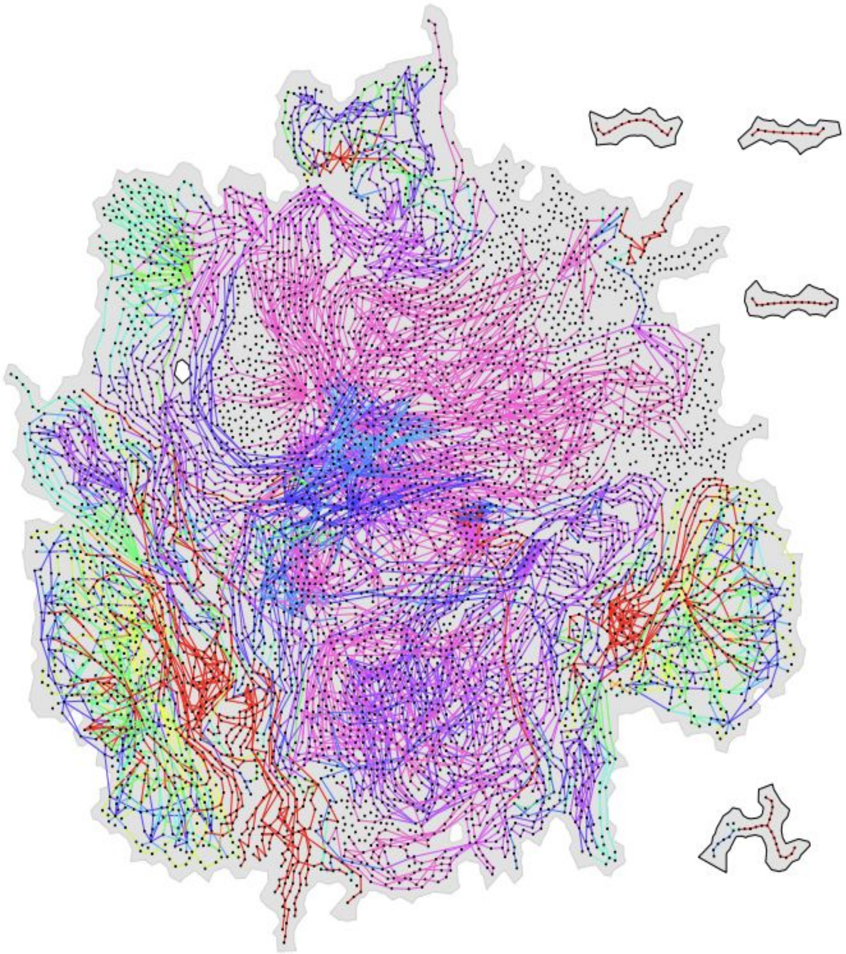


Fig. 1. Heatmap of messages sent during the ISCAS'89 s9234 simulation

Partitioning Strategy

- Use **METIS** graph partitioning tool
- Balance load (equal node weight)
- Minimize cross-partition edges (high edge weight = high comm.)

Random vs Profile-Guided

- Random: assigns objects arbitrarily → high traffic
- Profile-guided: groups high-comm. nodes → low traffic

Heatmap: Random Partitioning

- ~76% of messages cross partitions
- High network load, poor performance

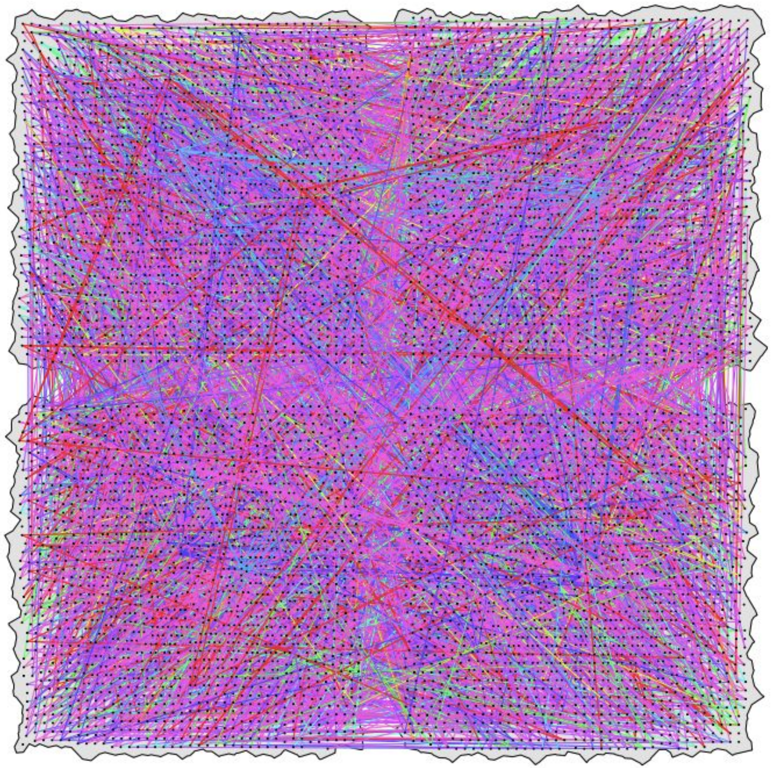


Fig. 3. Heatmap of messages sent during the ISCAS'89 s9234 simulation, partitioned randomly into four partitions

Heatmap: Profile-Guided

- Only ~1.4% of messages cross partitions
- Significant reduction in traffic

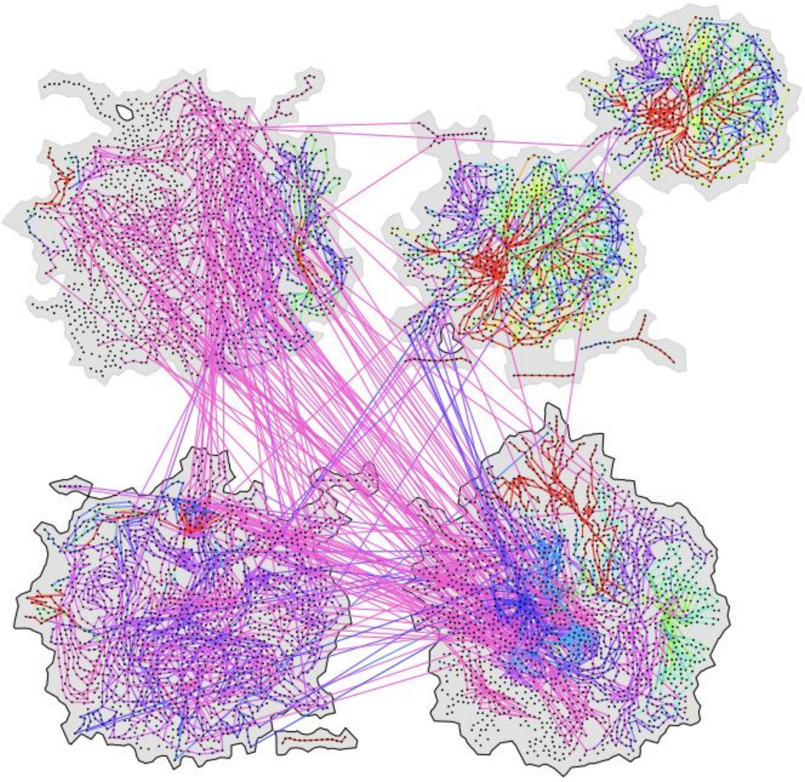


Fig. 4. Heatmap of messages sent during the ISCAS'89 s9234 simulation, partitioned into four partitions using the profile guided algorithm

Experimental Setup

- Beowulf-style compute cluster
- 2.33GHz Intel Xeon (quad-core, hyper-threaded)
- Each processor is hyper-threaded and runs a standard Linux environment.
- For inter-process communication, we used **OpenMPI**.
- 2, 4, and 8 node configurations

Benchmarks Used

- ISCAS’89 digital circuits
- RAID storage controller simulation
- Mix of structured and irregular models

ISCAS s5378

- Medium-size circuit
- Profile-guided speedups: 2.11x – 2.51x

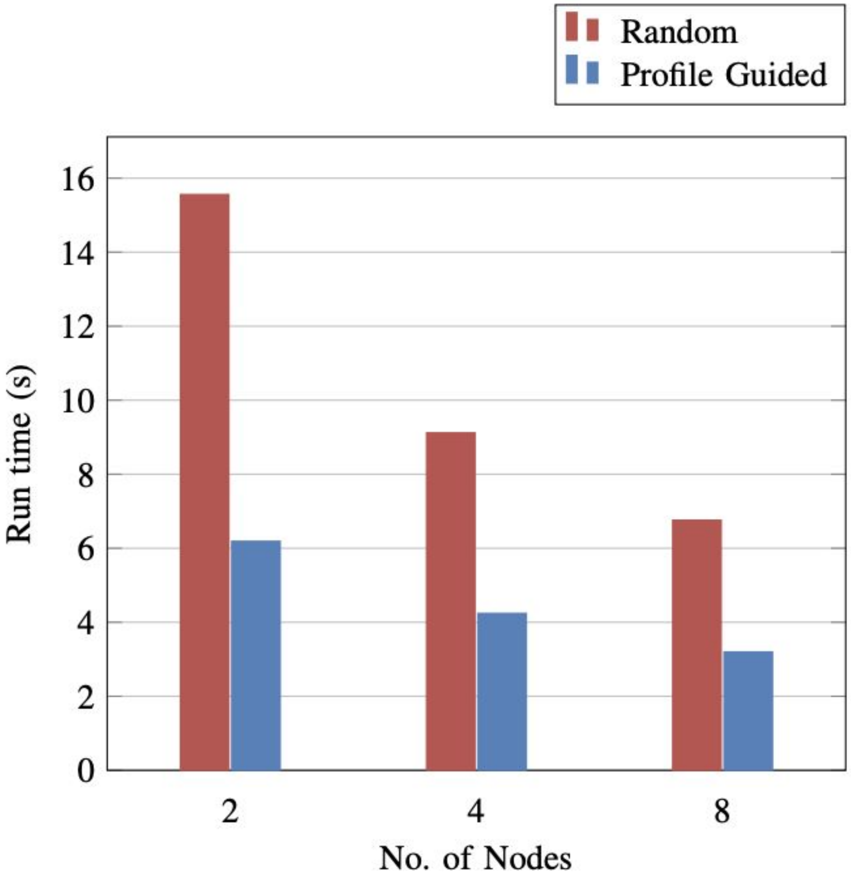


Fig. 7. Run time of ISCAS'89 simulation using the s5378 circuit

ISCAS s9234

- Dense, high-comm. circuit
- Speedups up to 6.04x
- Drop in perform as partitions increase

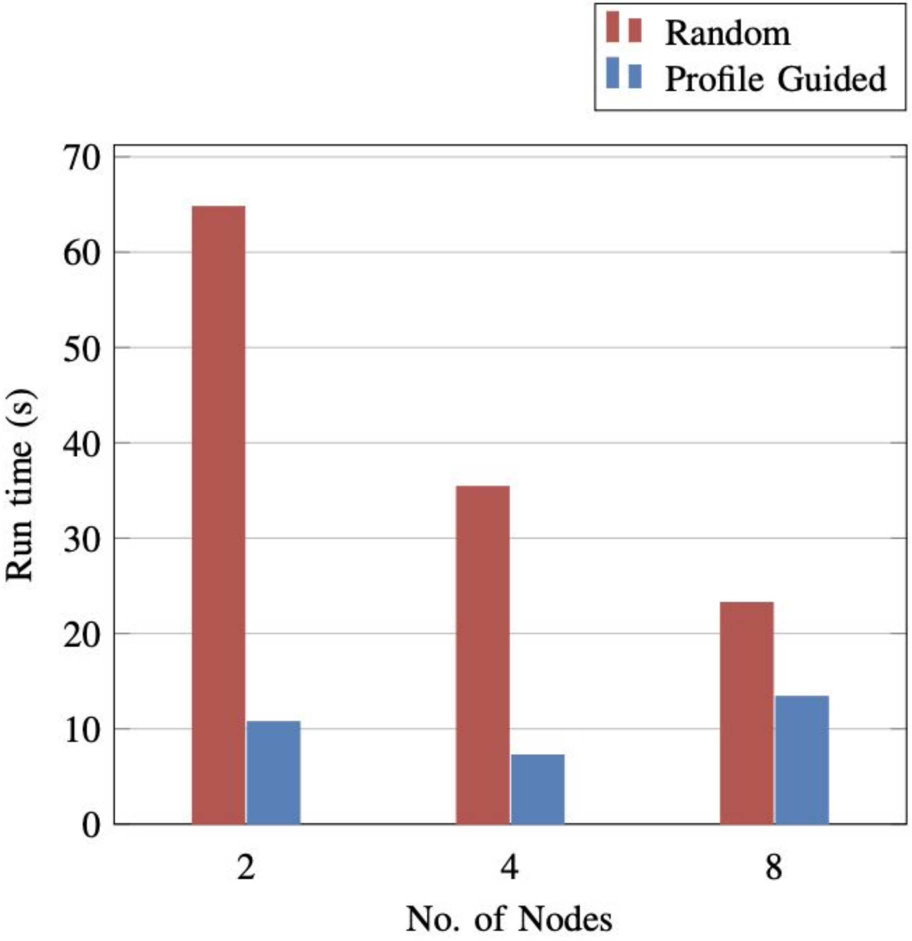


Fig. 8. Run time of ISCAS'89 simulation using the s9234 circuit

ISCAS s38584.1

- Largest circuit
- Complex comm. structure
- Better scalability at higher partition counts

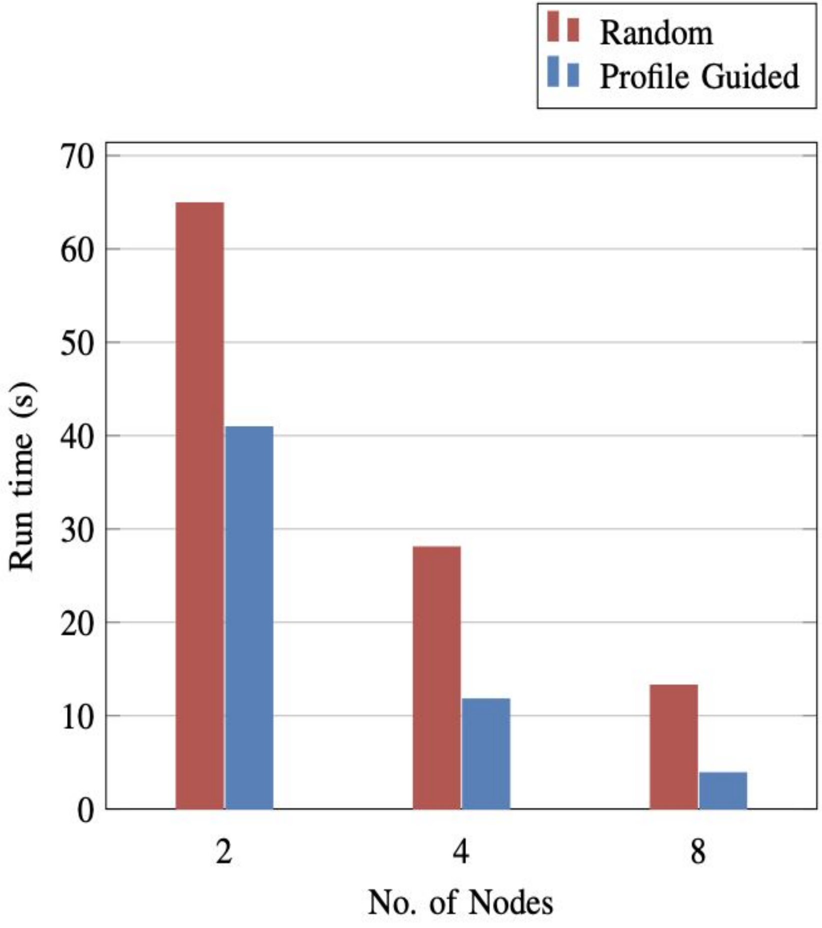


Fig. 9. Run time of ISCAS'89 simulation using the s38584.1 circuit

RAID Model

- 32 disks, 8 controllers, 96 I/O generators
- Structured communication
- Consistent 5× speedup

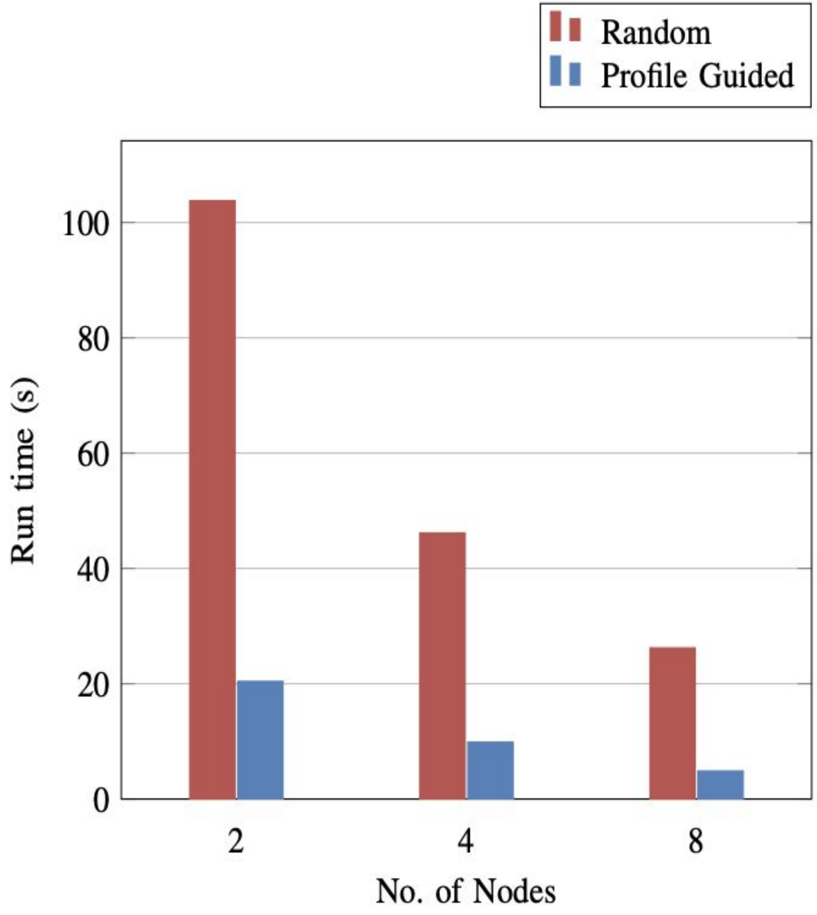


Fig. 6. Run time of RAID simulation

Summary of Results

Model	No. of Nodes	Random	Profile Guided	Speedup
ISCAS89: s5378	2	15.57	6.19	2.51×
	4	9.12	4.25	2.15×
	8	6.77	3.20	2.11×
ISCAS89: s9234	2	64.76	10.73	6.04×
	4	35.39	7.23	4.89×
	8	23.23	13.37	1.74×
ISCAS89: s38584.1	2	64.92	40.92	1.59×
	4	28.05	11.76	2.39×
	8	13.25	3.87	3.42×
RAID	2	103.79	20.41	5.09×
	4	46.16	9.78	4.72×
	8	26.22	4.86	5.40×

Trend Observations

- Profile-guided consistently outperforms random
- Models with regular structure (e.g., RAID) benefit more
- Partitioning gets harder with more nodes

Why It Works

- Reduces network load
- Improves synchronization
- Maintains load balance

Future Work

- Dynamic partitioning during runtime
- Combine with application-specific heuristics
- Use machine learning to predict partitions

Broader Implications

- Energy-efficient simulations
- Scalable to 1000s of nodes
- Generalizable beyond DES

Key Takeaways

- Communication dominates in PDES performance
- Intelligent partitioning = critical
- Profile-guided method shows real-world impact

Appendix A – Circuit Specs

Circuit	Flip-Flops	Inverters	Gates
s5378	179	1775	1004
s9234	228	3570	2027
s38584.1	1426	7805	11448

Appendix B – System Details

- Quad-core Xeon, 2.33GHz, hyper-threaded
- OpenMPI cluster, METIS v5
- Profiling scripts written in C++

Thank You!

Got questions? Let’s discuss!

Sunil Reddy Maram

smaram1370@gmail.com