Unified Performance Modeling Stack for Distributed GPU Applications: Complementing Analytical Insights with Machine Learning

Urvij Saroliya[1], Eishi Arima[1](Advisor), Martin Kronbichler[2](Advisor)
[1]Chair of Computer Architecture and Parallel Systems, Technical University of Munich, Germany
[2]Faculty of Mathematics, Ruhr University Bochum, Germany

1. INTRODUCTION

⚙️ **Modular Integrations.** 🎯 **Delivering Accuracy.** 💡 **Uncovering Insights.**

"Enabling Performance Modeling for Scalable GPU Computing"

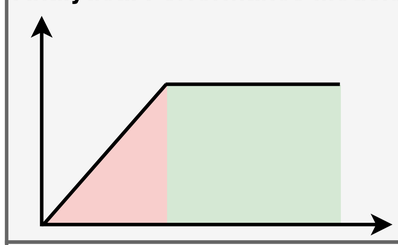
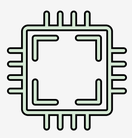
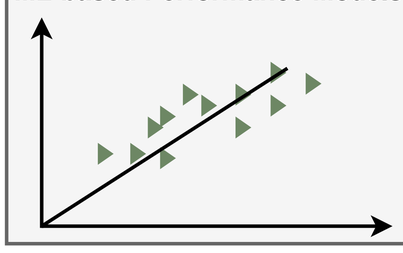
BACKGROUND

- Executing applications on large-scale GPU systems is often costly and time consuming.
- Without committing compute resources, performance modeling enables to
  → ⏱️ **estimate runtime**, 📈 **evaluate scalability**, and ⏳ **identify bottlenecks**.

CHALLENGES

- ➢ Accurate analytical models requires ***in-depth considerations*** of the architecture.
- ➢ Using performance data, ML models performance well but ***sacrifices interpretability***.
- ➢ Existing tools target specific GPU performance tasks:
  → ***no unified ecosystem*** for large-scale GPU performance modeling.

GOAL: Build a software stack for modeling and analyzing distributed GPU performance.

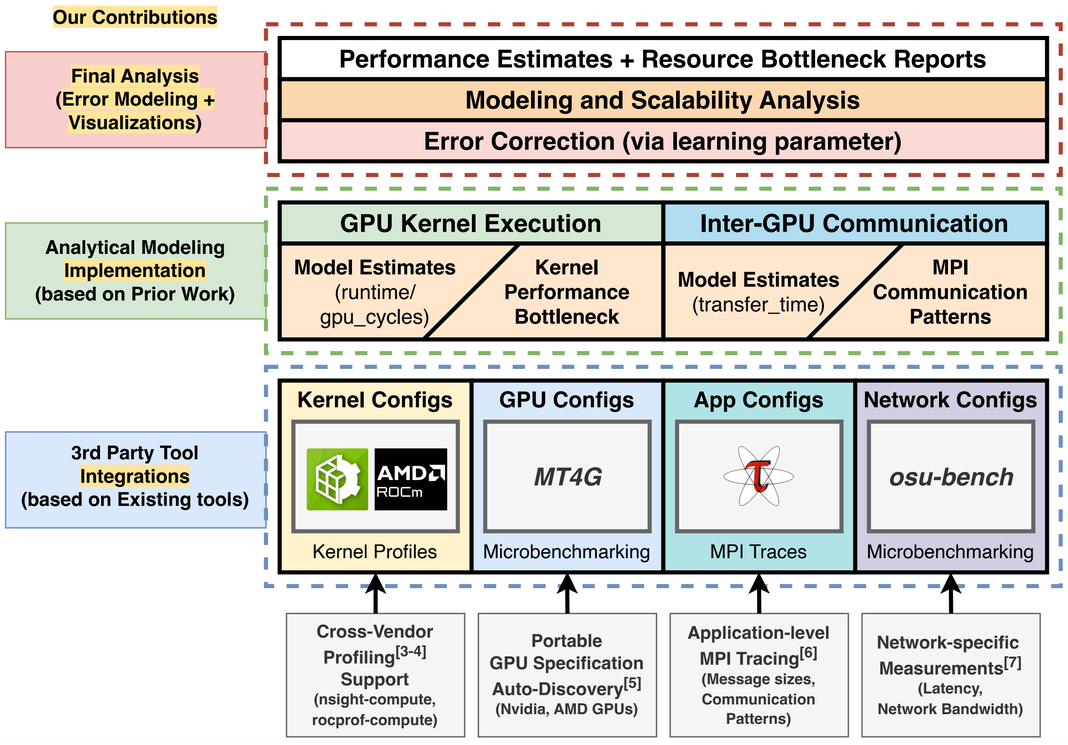
Analytical Performance Models 	 "White-box": Interpretable Results	 Detailed Architecture Insight Required	 Accuracy Bound to Analytical Assumptions
ML-based Performance Models 	 "Black-box": Insights Unavailable	 Data-driven: No Analytical Basis Needed	 High Prediction Accuracy: Curve Alignment

CONTRIBUTIONS

- ❖ ***Interpretability of Results*** ✅ → insights using "analytical machine model"
- ❖ ***Estimation Accuracy*** ✅ → via ML-based error correction
- ❖ ***Portability*** ✅ → enabling "cross-vendor/platform analysis" (Section 2)
- ❖ Performed evaluations using two real-life applications:
  (1) Helmholtz equation solver from Computational Fluid Dynamics (CFD), and
  (2) Gromacs application from Molecular Dynamics (MD).

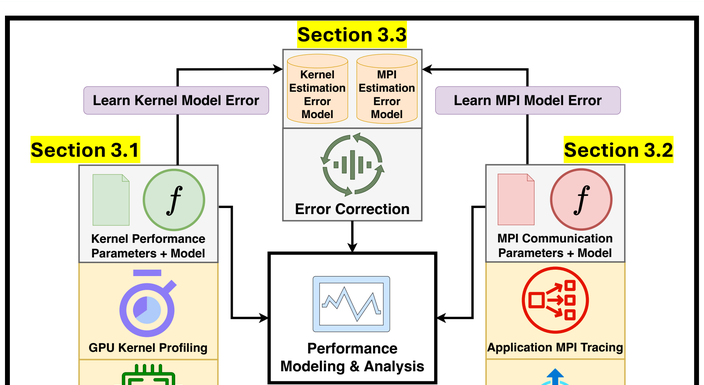
2. UNIFIED SOFTWARE STACK DESIGN

2.1 Implementation Overview



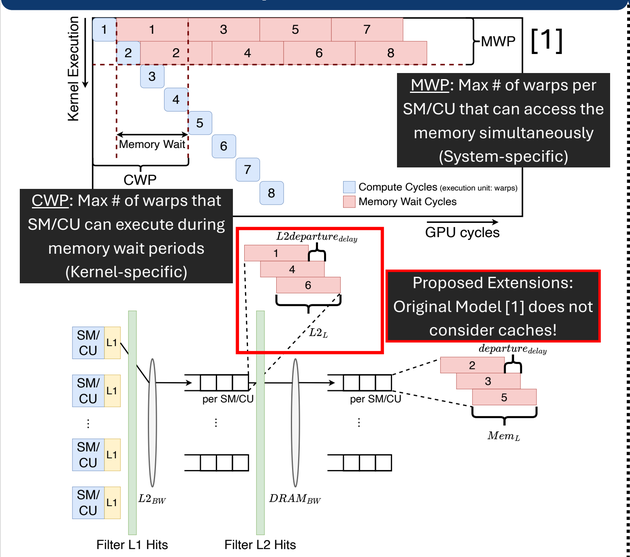
2.2 Proposed Workflow

- Integration of various components and performance tools.
- **Modular Approach**
  → allows use of other analytical/ML models.
- Two candidate models investigated for performance analysis. (Section 3)



3. PERFORMANCE MODELS

3.1 GPU Warp Parallelism Model



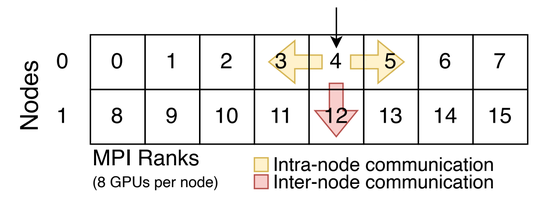
Using **an abstract machine model for GPU**
→ helpful for kernel analysis!

3.2 MPI Communication K-Model

[2]

$$T = \alpha + \frac{K_{inter}}{K_{total}} \cdot k \cdot n \cdot \beta$$

$k = 8, K_{inter} = 1, K_{total} = 3$



- α → start-up time
- β → (network bandwidth)$^{-1}$
- n → message size
- k → # of processes per node
- K_{inter} → max. # of inter-node messages
- K_{total} → max. # of total messages

Explicitly **distinguish b/w intra/inter-node**
→ important for multi-GPU scenario!

3.3 ML-based Error Correction Model

GIVEN

Performance data → X
Predicted runtime → $m(x)$
(Analytical Model)
Actual runtime → t

GOAL

Learn error function: $e(x)$,
→ for getting corrected runtime prediction $\hat{t}$

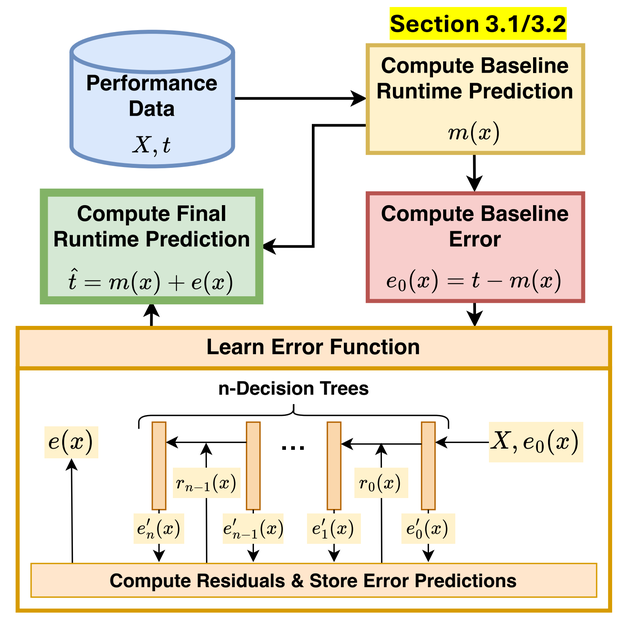
$$\hat{t} = m(x) + e(x)$$

METHODOLOGY

Use of Gradient Boosting Regression (GBR)
→ a stage-wise ensemble of decision trees.

WHY GBR?

Non-linear error patterns can be well-identified for underlying analytical models.

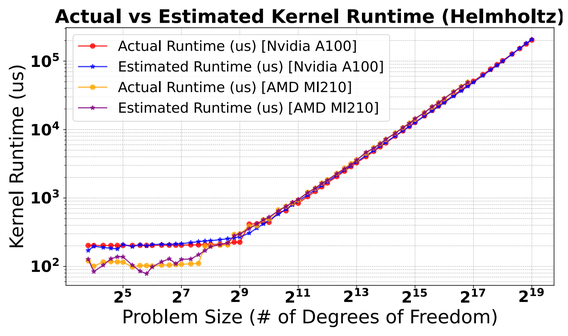


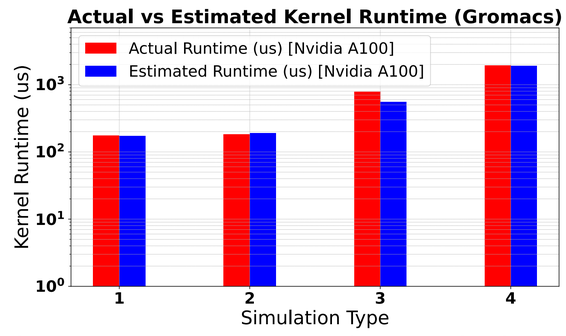
4. EXPERIMENTAL RESULTS

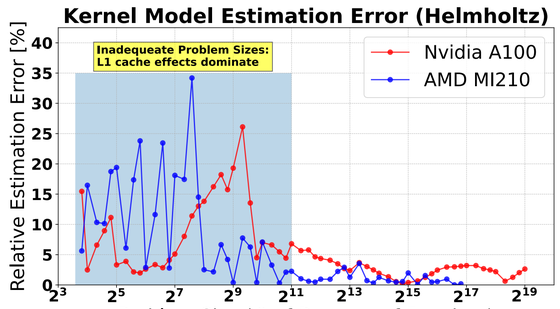
4.1 Setup

Setup 1 (Nvidia GPU)	8 x NVIDIA A100 40GB HBM2e 400W TDP 2 x AMD EPYC 7713 (Zen3), 2 x 64 cores @2.0 GHz **(Multiple-GPUs/nodes: up to 2 x 8 = 16 GPUs)**	**Application 1**	**PDE Solver – Helmholtz Equation:** High-order FEM with Polynomial Degree 7 using deal.II Library [8], Varying # of Degrees of Freedom that fits on the GPU
Setup 2 (AMD GPU)	AMD Instinct MI210 64GB HBM2e 300W TDP 2 x AMD EPYC 7713 (Zen3), 2 x 64 cores @2.0 GHz **(Single-GPU/node)**	**Application 2**	**Molecular Dynamics Simulations – Gromacs** [9]**:** Simulation Types: **(1)** R-143a in Hexane, **(2)** RNA Piece with Explicit Water, **(3)** Protein inside a Membrane, **(4)** Protein in Explicit Water

4.2 Evaluations

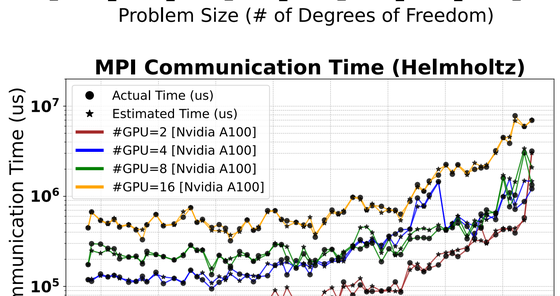






GPU KERNEL RUNTIME ESTIMATIONS
(Single GPU evaluations)

- High accuracy achieved for larger grid sizes for the Helmholtz Equation Solver:
  → relative estimation error ***below 5%*** ***(up to 30% w/o error correction model)***.
- For MD problems using Gromacs:
  → relative estimation error under 3% in all but one case.



MPI DATA TRANSFER TIME ESTIMATIONS
(Multiple-GPUs/nodes evaluations)

- Strong and weak scaling for the Helmholtz Equation Solver:
  → average estimation error of ***8.52%*** ***(28.67% w/o error correction model)***.
- ***Larger-scale evaluations planned!***

CONCLUSION: ***Using a combination of analytical insights and ML-based error corrections, this work proposes software ecosystem for end-to-end performance modeling and scalability analysis for distributed GPU applications, achieving estimation errors < 5% for GPU kernels (for adequate problem sizes) and < 9% for inter-GPU communications.***
FUTURE WORK: ***(1) validate model for other HPC and AI workloads, (2) compare against other SOTA performance modeling methods or tools.***

WATCH POSTER VOICE-OVER:

REFERENCES:
[1] Sunpyo Hong et al. 2009. An analytical model for a GPU architecture with memory-level and thread-level parallelism awareness. In Proceedings of the 36th annual international symposium on Computer architecture. 152–163.
[2] Jaemin Choi et al. 2020. End-to-end performance modeling of distributed GPU applications. In Proceedings of the 34th ACM International Conference on Supercomputing. 1–12.
[3] NVIDIA Corporation. 2025. Nsight Compute Documentation - NsightCompute 12.9 documentation 2025. NVIDIA Corporation. https://docs.nvidia.com/nsightcompute/ Accessed: 2025-08-14.
[4] Xiaomin Lu et al. 2025. ROCm/rocprofiler-compute: v3.1.0 (12 February 2025). https://doi.org/10.5281/zenodo.7314631
[5] Stepan Vanecek et al. 2025. MT4G: A Tool for Reliable Auto-Discovery of NVIDIA and AMD GPU Compute and Memory Topologies. In Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis.
[6] Sameer S Shende et al. 2006. The TAU parallel performance system. The International Journal of High Performance Computing Applications 20, 2 (2006), 287–311.
[7] Ohio State University. 2025. MVAPICH :: Benchmarks – OSU Micro-Benchmarks 7.5.1. https://mvapich.cse.ohio-state.edu/benchmarks/ Accessed 2025-08-14.
[8] deal.II. 2025. The deal.II Finite Element Library. https://dealii.org/ Accessed: 2025-09-28.
[9] Gromacs. 2025. Welcome to GROMACS. https://www.gromacs.org/ Accessed: 2025-09-28.

ACKNOWLEDGEMENT
This work was conducted under the doctoral supervision of Professor Martin Schulz, TU Munich. It has been supported by the German Federal Ministry of Research, Technology, and Space (BMFTR) through the SCALEXA initiative and the PDExa project (16ME0641). Special thanks extended to the Erlangen National High Performance Computing Center (NHR@FAU, Germany) and Leibniz Supercomputing Centre (LRZ, Germany), for providing access to the compute resources.

With funding from the: Federal Ministry of Research, Technology and Space

SC25 St. Louis, hpc HQ ignites.