GenASPEN: Agentic AI for Automatic Performance Modeling of HPC Codes

Grace McLewee‡*, Mohammad Alaul Haque Monil†*,
Pedro Valero-Lara†, Narasinga Rao Miniskar†, William Godoy†, Allen Malony‡, Keita Teranishi†, Jeffrey S. Vetter†.
‡University of Oregon, †Oak Ridge National Laboratory, USA. [*co-first authors], contact: monilm@ornl.gov

ASPEN ORNL repo

GenASPEN ORNL repo

Introduction and Motivation

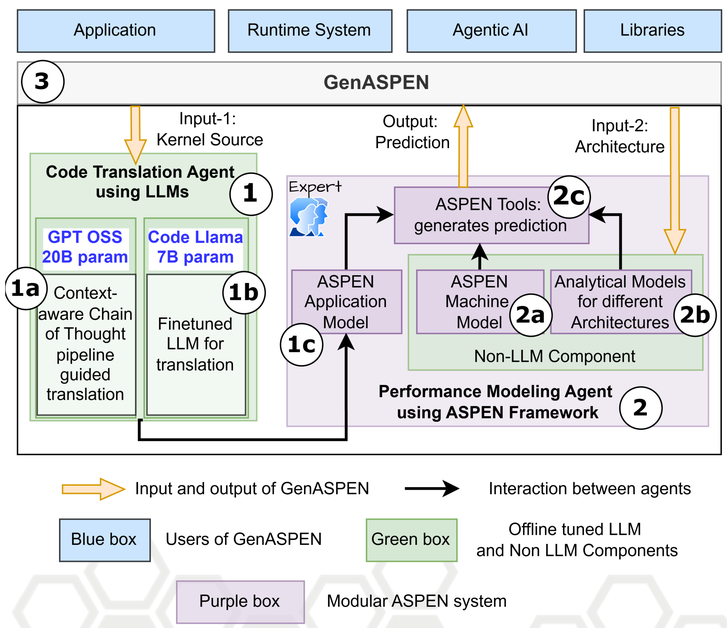
- Heterogeneous systems with GPUs make automated performance modeling both more essential and more difficult
- Compiler-based frameworks are accurate but must be re-integrated for every new architecture; frontier LLMs are flexible but predict runtime poorly
- Custom LLMs buy accuracy at prohibitive cost — Omniwise [4] needed 955,000 profiled kernels

Why Agentic AI for ASPEN

- ASPEN decouples an abstract application model, a machine model, and prediction tools invoking analytical models [2]
- Two steps stayed manual: translating source into an application model, and choosing the analytical model
- LLMs are strong at code understanding, analytical models at quantitative prediction — give each task to the component best equipped for it.
- Agentic AI can orchestrate LLMs, apply reasoning and call non-LLM tools

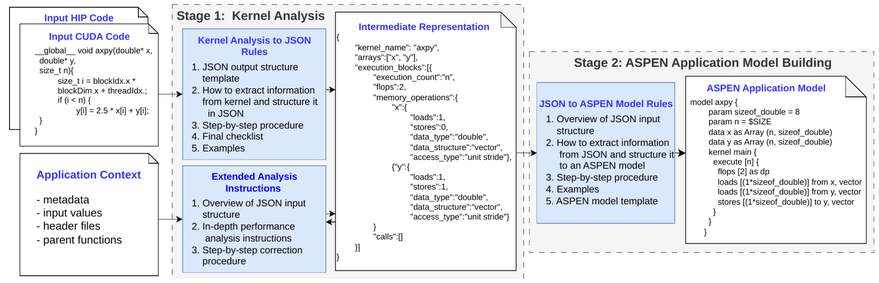
GenASPEN: Design and Flow

- Two Agents: Code Translation Agent and Performance Modeling Agent
- ③ API takes the kernel source and the target architecture
- ① Code Translation Agent: Either ①a a context-aware chain-of-thought pipeline on GPT-OSS 120B, or ①b a Code Llama 7B assistant fine-tuned with ChatHPC [3] — both emit ①c an ASPEN application model
- ② Performance Modeling Agent: ②c ASPEN tools bound to GPT-OSS 20B traverse ①c and ②a the machine model, then invoke ②b the analytical model for that GPU
- ②a and ②b are calibrated once per architecture, offline; experts fix corner cases and saved models bypass every LLM call



Code Translation Agent (Fine tuning vs. CoT)

Context Aware Chain of Thought Pipeline (CA-CoT)



Fine-tuning is up to 101× faster on as little as 1.74% of the tokens, but breaks down on the three proxy applications; CA-CoT translated every one correctly, with no expert intervention.

Two translation strategies: speed vs. robustness

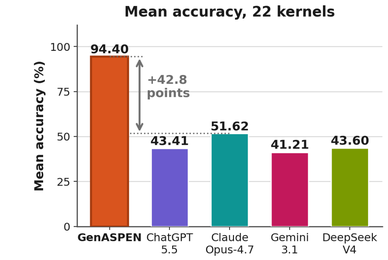
Operations and Execution time Prediction

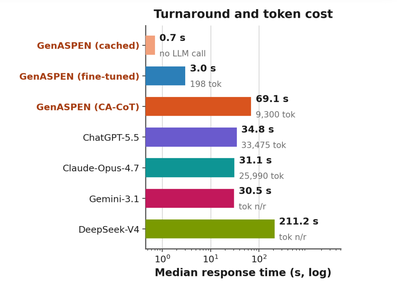
Operation counts 98.5% (FLOPs) / 98.375% (MOPs) • **Execution time 94.40% (CUDA) / 93.09% (HIP)**

- Total 22 benchmarks and applications evaluated on NVIDIA H100 and AMD MI300A
- Operation counts are exact on 19 of 22 benchmarks (98.50% FLOPs, 98.375% MOPs) because the LLM resolves both the computational pattern and its structural parameters.
- Execution time is predicted to 94.40% accuracy on the H100 and 93.09% on the MI300A.
- Residual error sits in the analytical model rather than the application model: MiniMDock is under-predicted because synchronization is not yet represented.

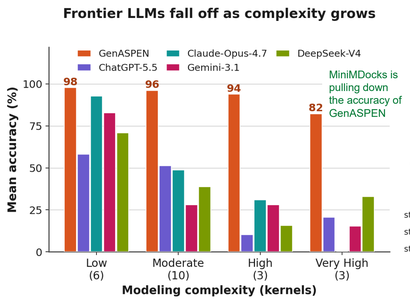
Frontier LLMs vs. GenASPEN

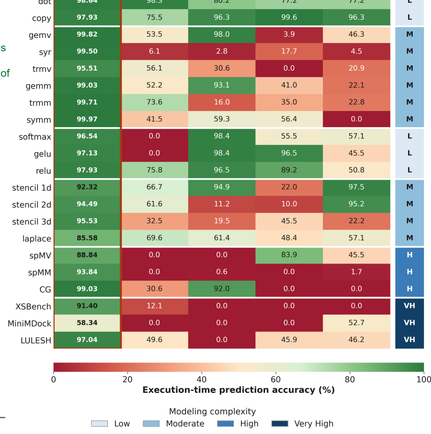
Accuracy and Cost per Prediction



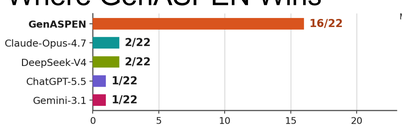


Accuracy by Complexity and Kernel



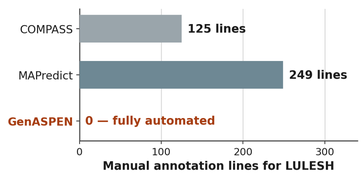


Where GenASPEN Wins



Classic Approaches vs. GenASPEN

Manual Effort to Model LULESH



COMPASS [ICS'15] needs up to 125 annotation lines and MAPredict [1] up to 249; GenASPEN needs none.

LLMPerf predicts SpMV runtime at 32.5–56.8% MAPE; GenASPEN reaches 11.16% APE.

Conclusion & Future Work

- 94.40% (H100) and 93.09% (MI300A) mean execution-time accuracy
- Beats four frontier LLMs on 16 of 22 kernels, on a fraction of the tokens
- Zero manual annotation, where classic tools need hundreds of lines
- *Ongoing and Next:* Energy and communication modeling, agentic recalibration

Acknowledgments:
This work is funded by DOE ASCR AI for Science projects, Durban and Ellora.

References:
[1] Monil, M.A.H et. Al. "MAPredict: Static Analysis Driven Memory Access Prediction Framework for Modern CPUs," ISC 2022.
[2] Spafford and Vetter, "ASPEN: A domain specific language for performance modeling," SC12, 2012.
[3] Valero-Lara et al., "ChatHPC: Foundations for a productive and trustworthy AI-assisted HPC ecosystem," SC 2025.
[4] Wang et. Al., "Omniwise: Predicting gpu kernels performance with llms," arXiv preprint
[5] McLewee and Monil et al. "GenASPEN: An Agentic AI Framework for Automatic Performance Modeling and Prediction of HPC Codes. Paper accepted at Cluster 2026.