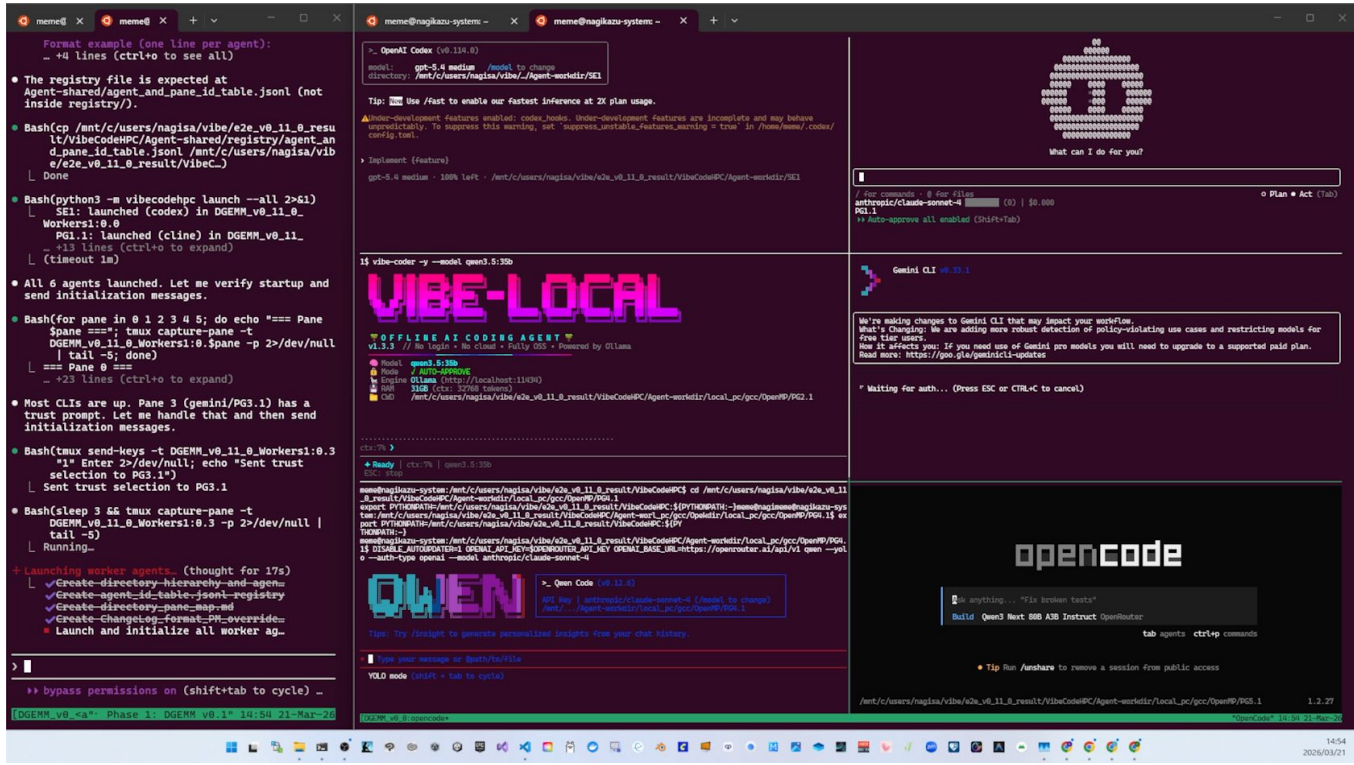
VibeCodeHPC

A Multi-Agent LLM Framework for Autonomous HPC Code Auto-Tuning

— Toward an Agent-Driven Foundation for Semiconductor Design Workloads

Shun-ichiro Hayashi

(Graduate School of Informatics, Nagoya University)



Workshop on LLM-Driven Code Generation and Automation for HPC and Scientific Computing

Monday, July 6, 2026　14:00〜14:25

LIP6, Sorbonne University, Paris

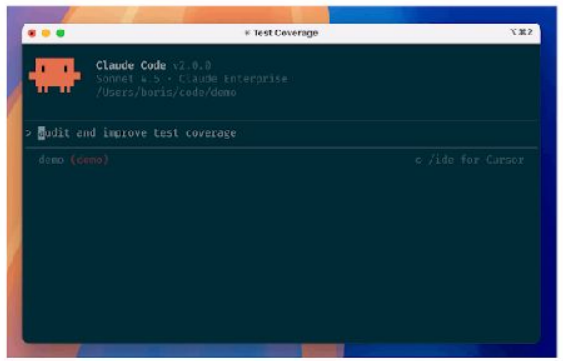
Motivation

HPC Programming Is Hard to Automate

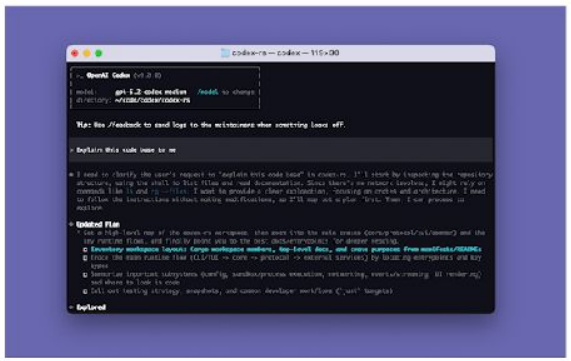
- HPC code is performance-critical, but the number of HPC experts is limited.
- Today's tuning pipeline is still mostly manual:
  - Diverse parallel programming models (MPI / OpenMP / CUDA / OpenACC)
  - Batch jobs, module systems, SSH access
  - Performance–accuracy–power trade-offs
- **Hardware / module information must be shared with the optimizer** — a persistent friction point.

→ How about a latest LLM running in a CLI agent?

A Single LLM Agent Isn't Enough



Claude Code

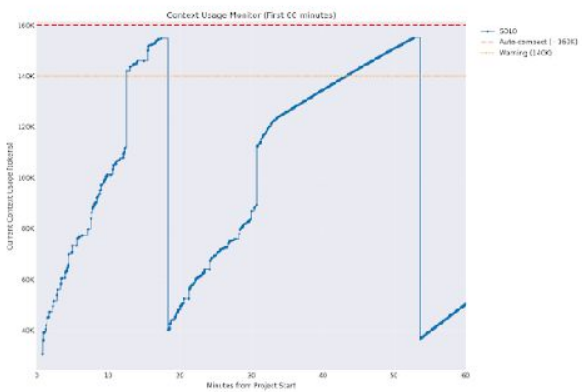


OpenAI Codex

LLM-based CLI coding agents are powerful — but used alone, they:

- exhaust their **context window** on long-running HPC workloads,
- **forget the spec** after auto-compact,
- and **call "done" too early**.

→ This gap is our target.



Solo: context hits auto-compact

Contributions

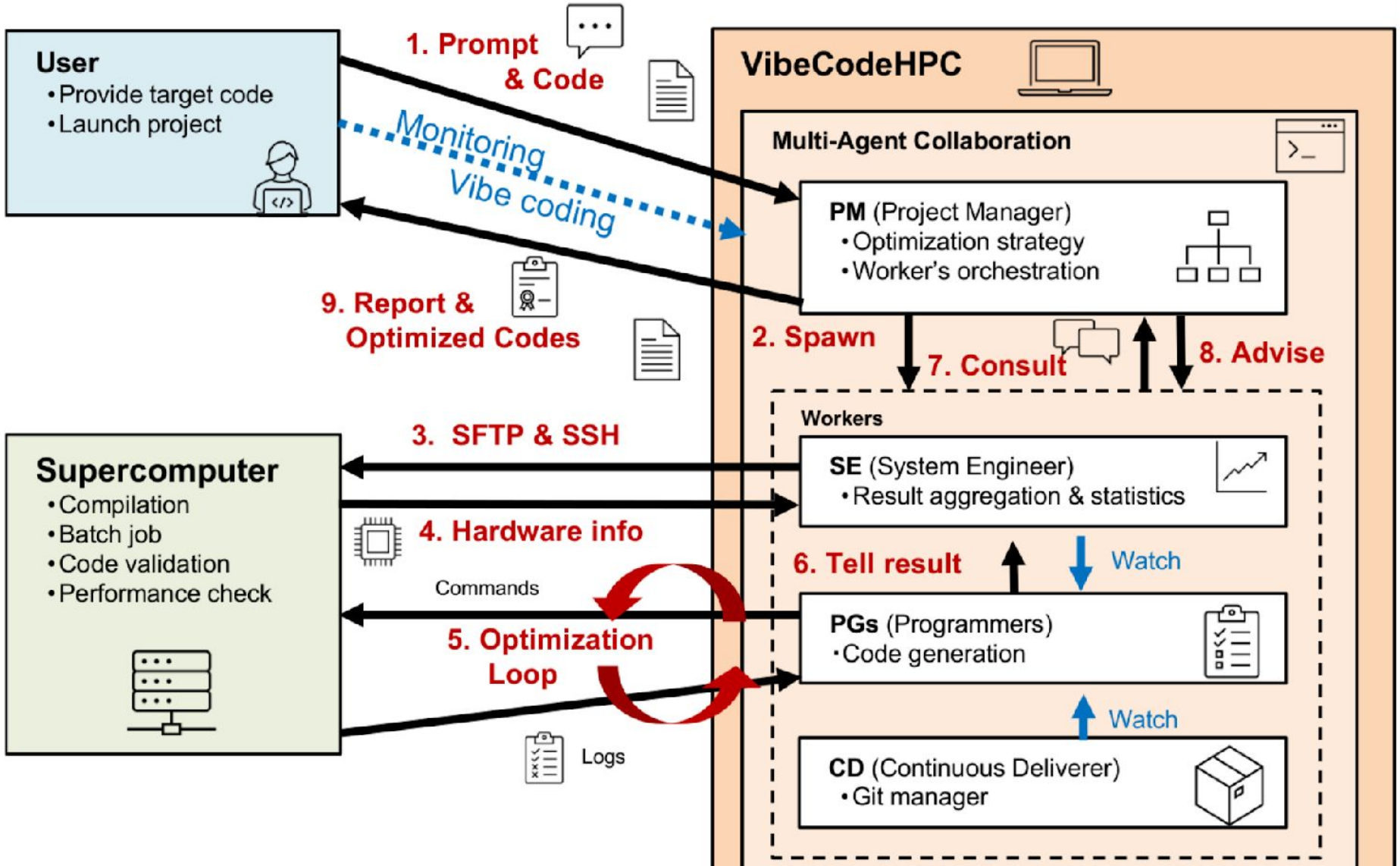
VibeCodeHPC: an open multi-LLM agent framework for HPC auto-tuning. *Backend in this experiment: Claude Code (Sonnet 4.5).*

1. **Role-specialized multi-agent collaboration**
   PM / SE / PG / CD, Tmux-based inter-agent comms, dynamic deployment *(implemented summer 2025, predating Anthropic's official sub-agents).*
2. **Long-term autonomous operation**
   Hook-based recovery, real-time context monitoring, budget tracking.
3. **Supercomputer-aware tooling**
   SSH/SFTP, batch jobs, modules, hardware/software discovery.

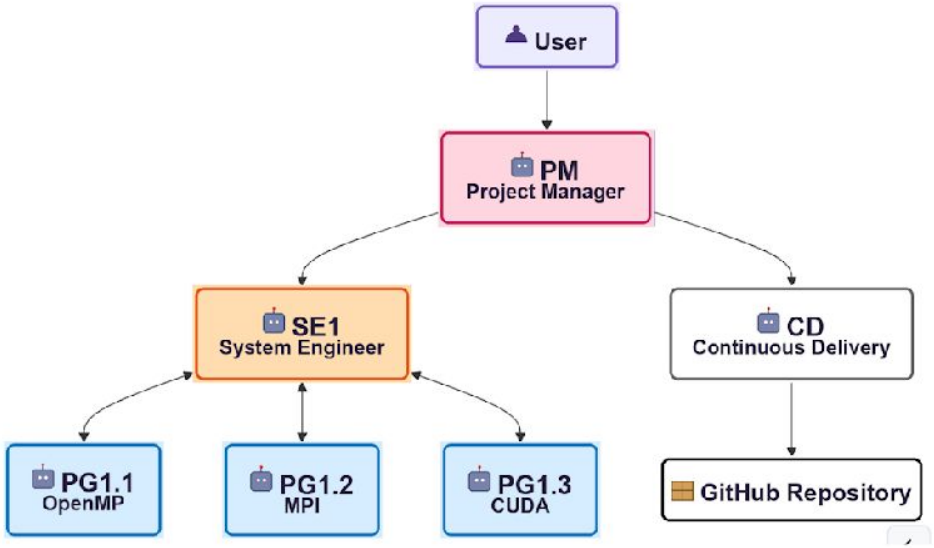
System Design

Workflow Overview

User hands the target code to VibeCodeHPC; Multi-Agent tunes the code on the supercomputer.



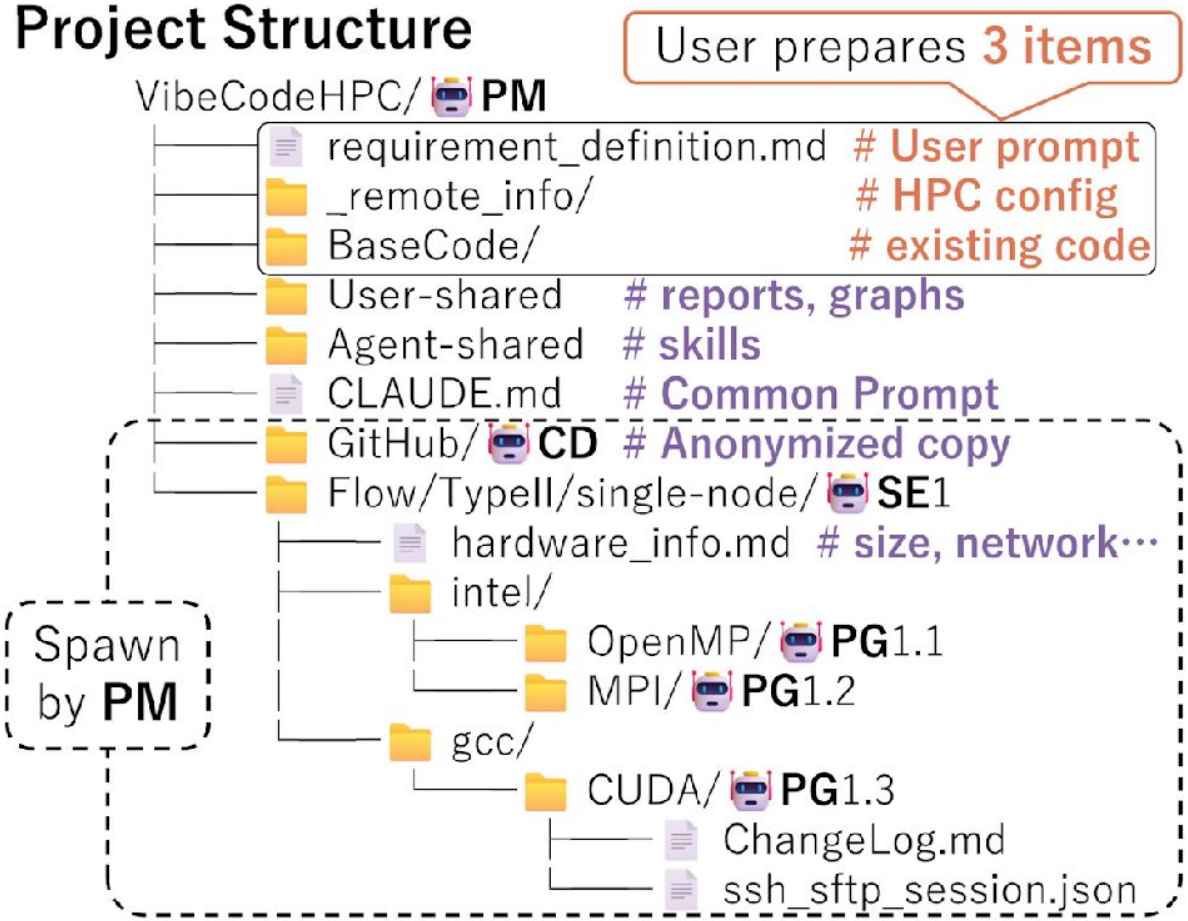
Multi-Agent Roles and Dynamic Organization



PM × 1	Orchestrates the project, talks with the user, **spawns agents dynamically**.
SE × 1–2	Manages PGs, aggregates results, inspects hardware and topology.
PG × many	Specialised per technology directory (e.g. OpenMP / MPI / CUDA).
CD × 1 (optional)	Anonymises secrets; pushes to GitHub.

3-tier hierarchy is preset; agents communicate asynchronously through ChangeLog + short messages.

Project Structure & Inputs



1. *`requirement_definition.md`*
   goals, hardware, constraints, 3-tier budget (min / target / upper)

2. *`_remote_info/`*
   SSH info, modules, point-usage script

3. *`BaseCode/`*
   the code to be optimized

Demonstration

Experimental Setup

Workload	Size			Time
Himeno (CFD)	SMALL: $129 \times 65 \times 65$	10 cores	V100×1	60–90 min
Matrix Multiplication	$n=2048$, no cuBLAS	40 cores	V100×4	45–60 min

full = 1 node = V100×4 + Xeon Gold 6230 (20 cores)×2

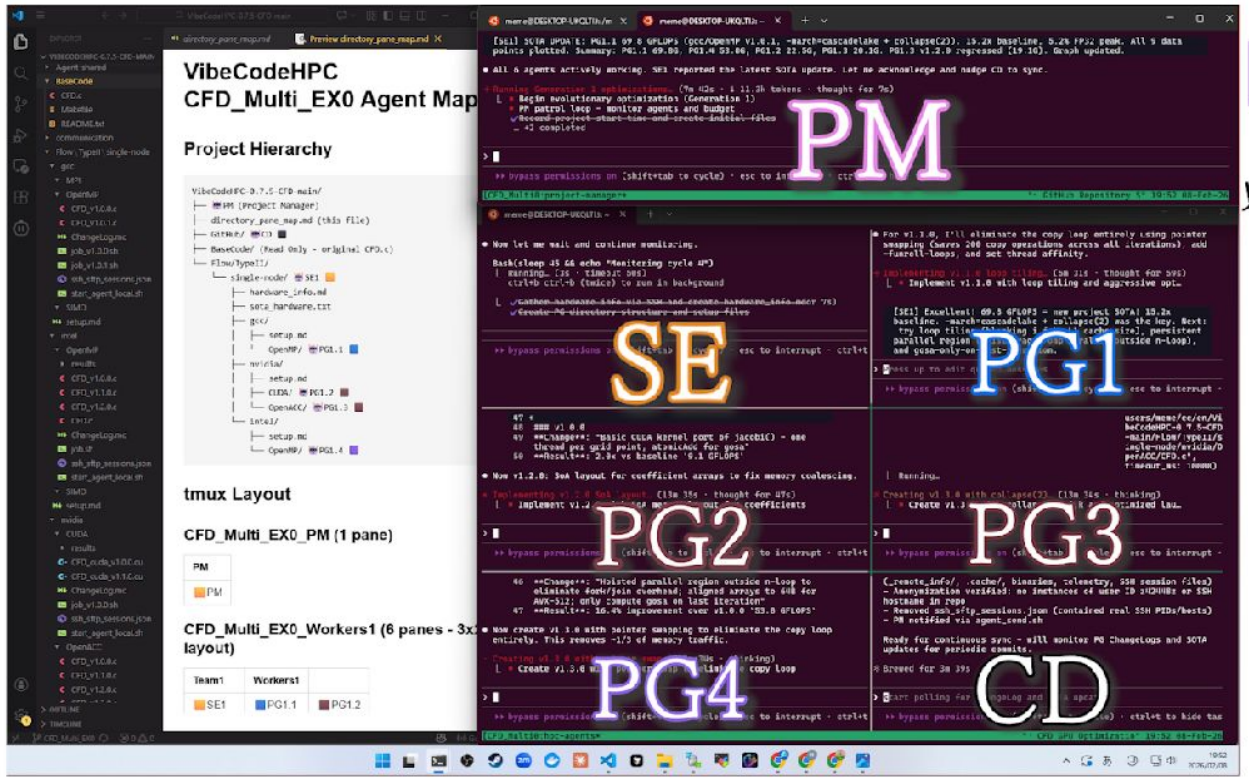
Agent Configurations (3 runs per config):

Multi = 1 PM + 1 SE + 4 PGs + 1 CD vs **Solo** = 1 agent doing all roles.

Metric: best GFLOPS reached within time budget (CPU↔GPU transfer included).

Local PC: WSL2, **Claude Code 2.0.8 (Sonnet 4.5)**, VibeCodeHPC v0.7.5.

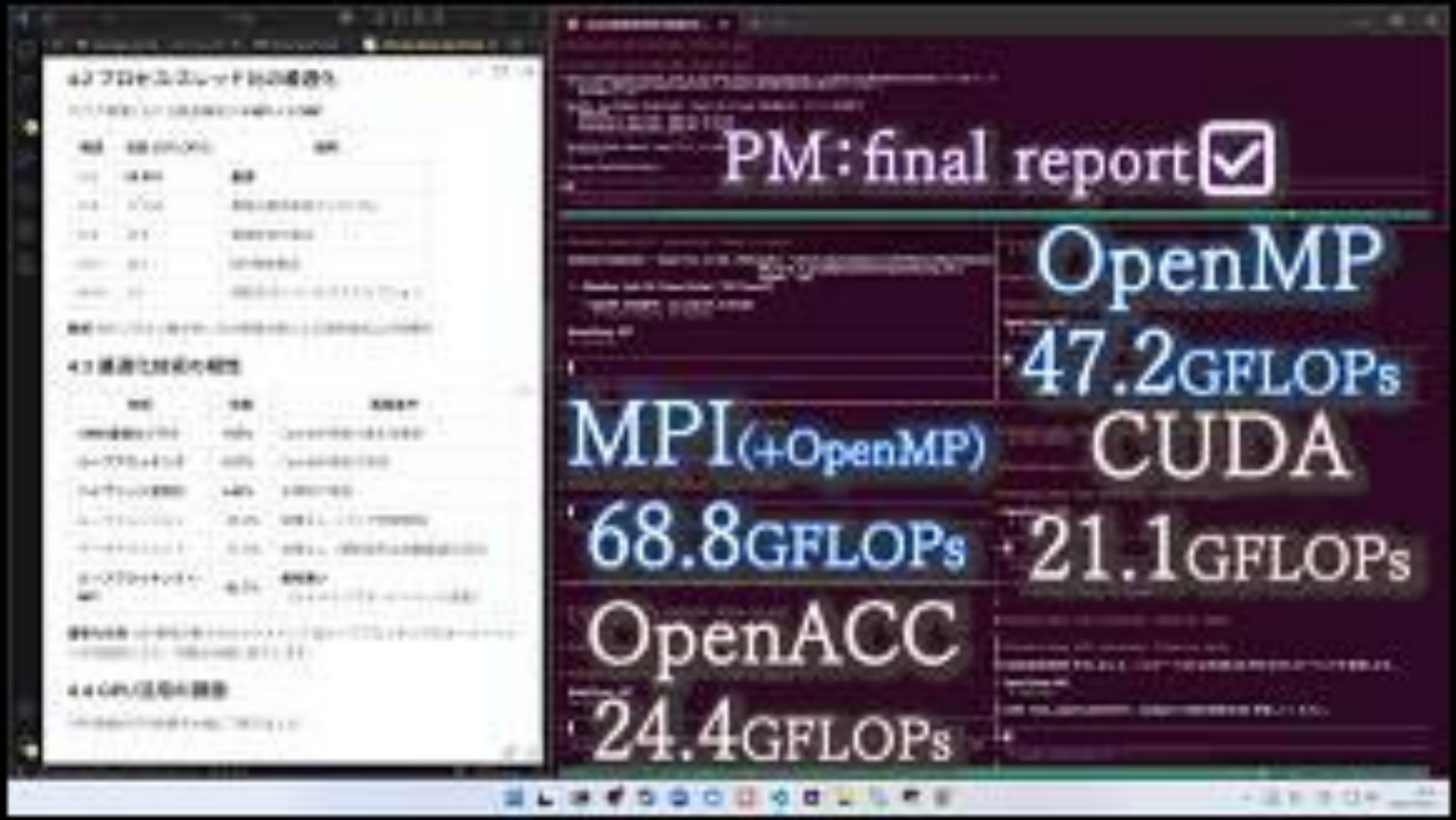
Live Demo: Himeno CFD with VibeCodeHPC



▶ demo

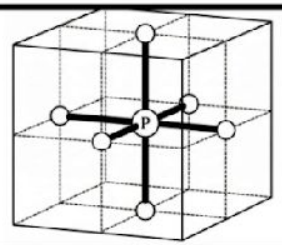
youtu.be/aN5UbA6ssQQ

- PM spawns SE / 4 PGs / CD on launch.
- PGs explore OpenMP, MPI, CUDA, OpenACC **concurrently**.
- SE aggregates GFLOPS from each *ChangeLog.md*.
- CD pushes artifacts to GitHub.


PM: final report
OpenMP
47.2GFLOPs
MPI(+OpenMP)
68.8GFLOPs
CUDA
21.1GFLOPs
OpenACC
24.4GFLOPs

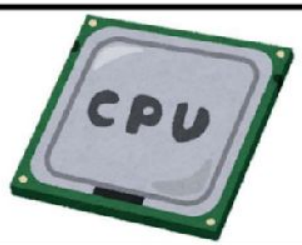
Evaluation

Aggregated Results: Two Workloads, Two Stories



$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

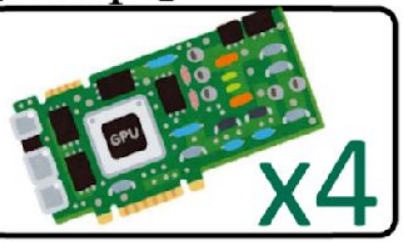
Himeno Benchmark
a 10-core CPU



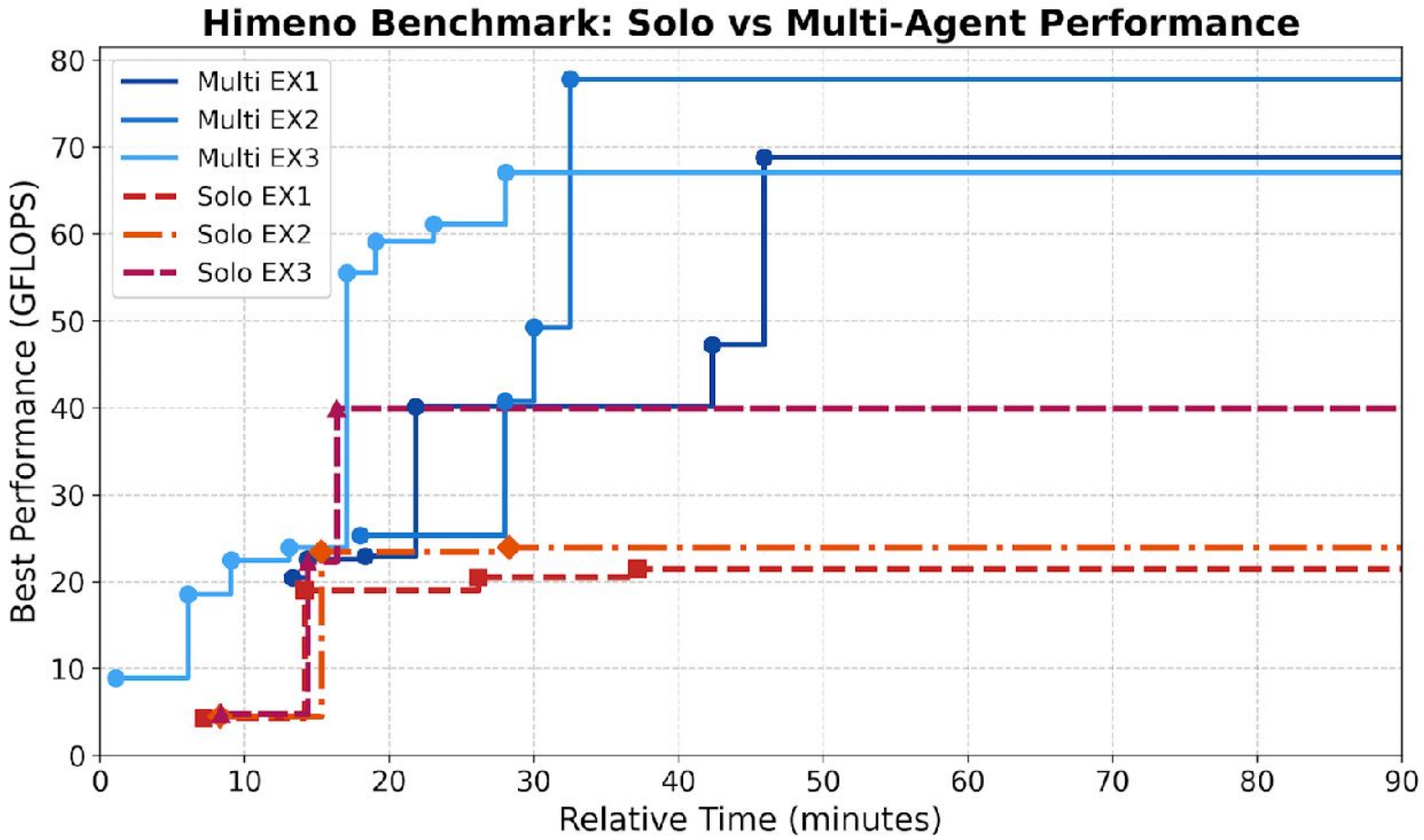
Target	Mode	EX1	EX2	EX3	Avg
CFD	Multi	68.8	77.8	67.1	**71.23**
	Solo	21.5	23.9	39.9	28.43
DGEMM	Multi	1385	3677	916	**1993**
	Solo	805	607	1098	837

[Gflops]

(A)(B)=(C) Matrix product
NVIDIA V100 GPUs



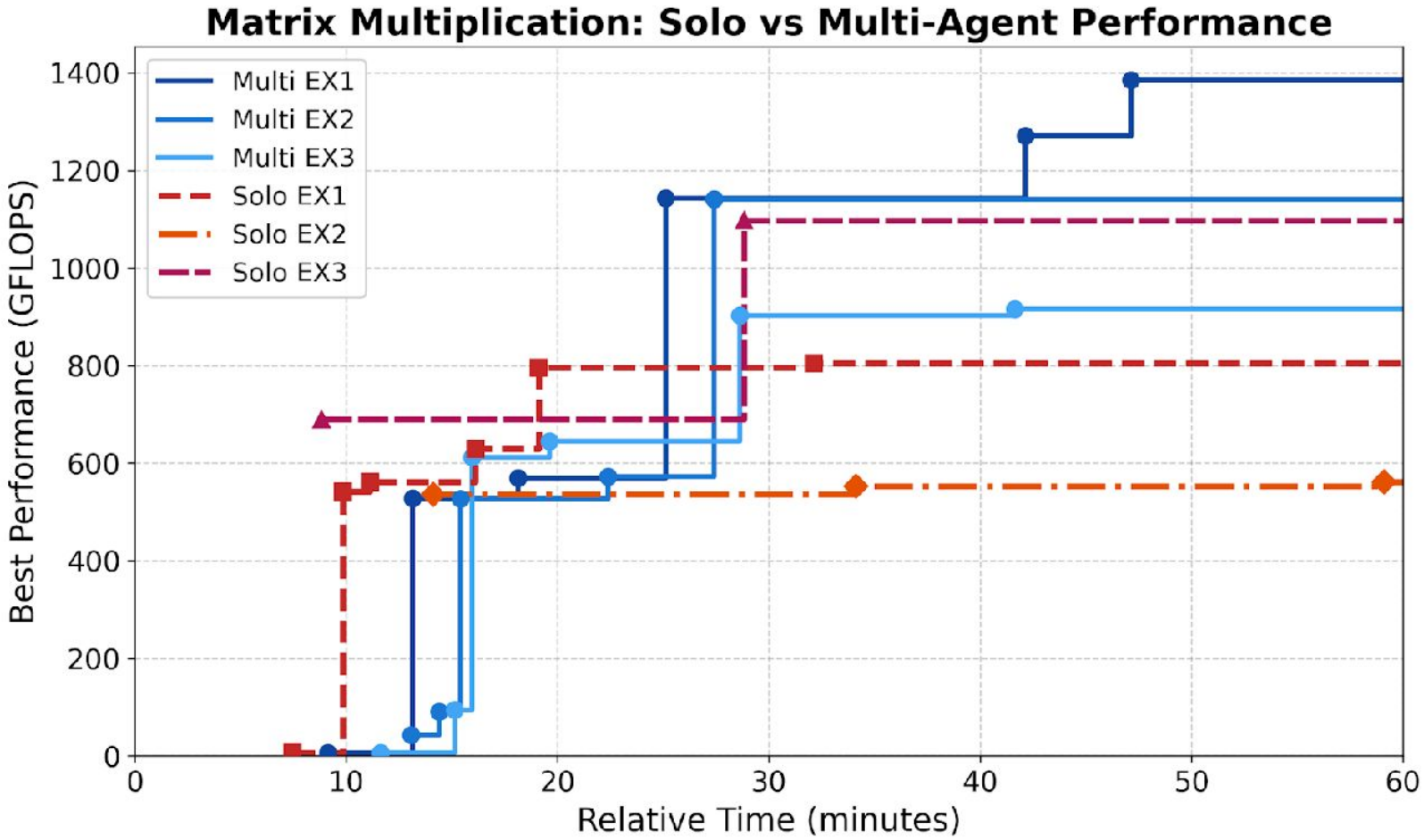
Himeno Benchmark (CFD): Solo vs Multi



Blue = Multi Red = Solo 3 runs each (EX1, EX2, EX3)

x: elapsed time y: GFLOPS (higher is better)

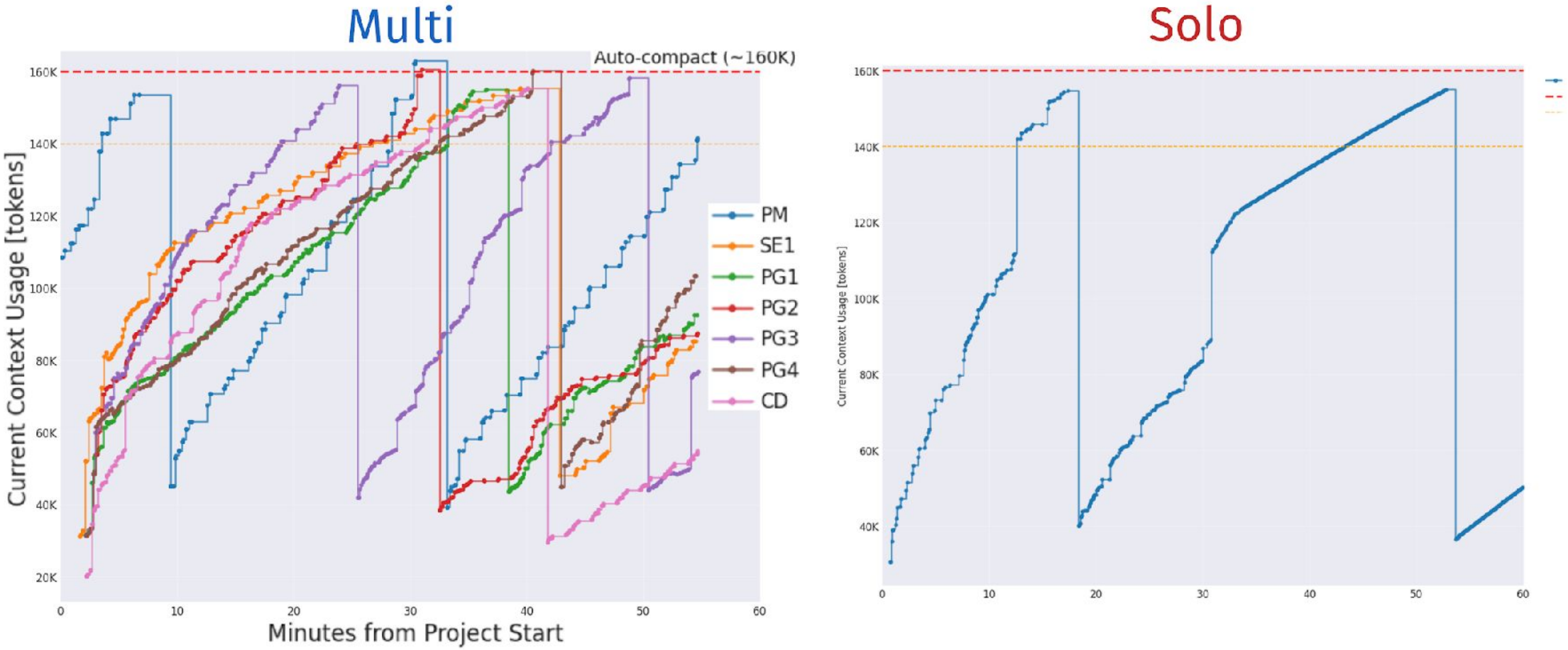
Matrix Multiplication: Solo vs Multi



Blue = Multi Red = Solo 3 runs each (EX1, EX2, EX3)

Discussion

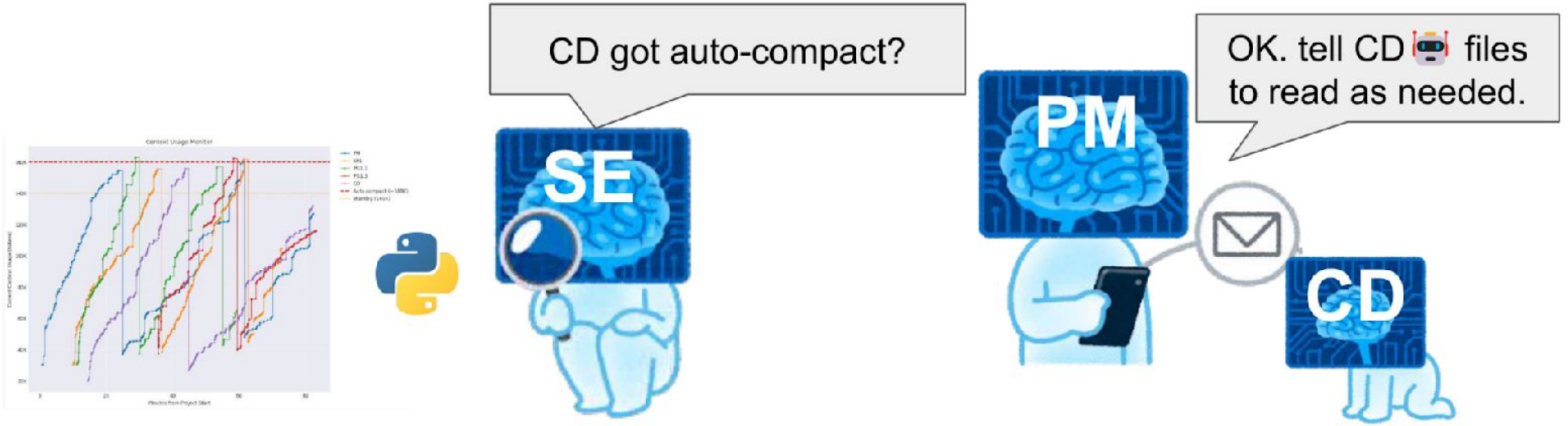
Context Usage: Multi vs Solo



- **Total memory pool grows** with agent count, so each PG can focus on coding.
- Other agents **rescue** a fellow agent when it hits auto-compact.

Surviving Auto-compact: Multi Agents Care for Each Other

Muti agents can take care of others after auto-compact

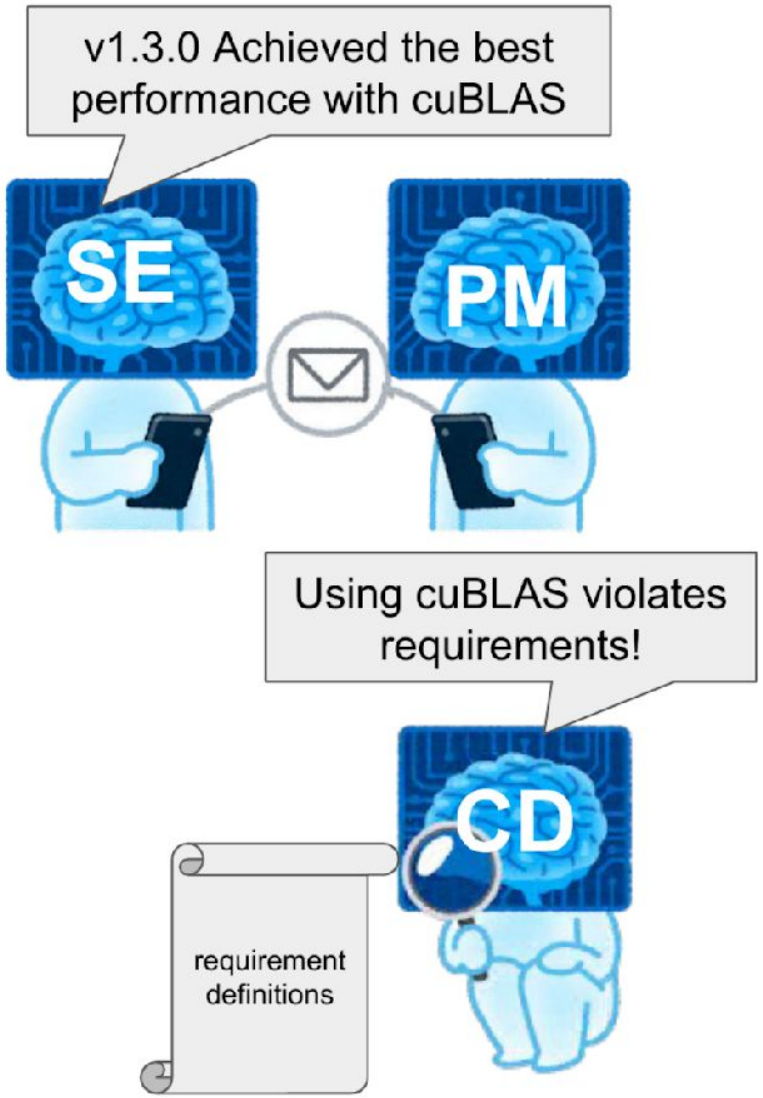


To SE1: "[PM Manager] PG2 successfully submitted Job ID: 2053140. budget_tracker recognizes 2 running instances (PG1 might have been submitted as well). Continuing to monitor CD context at 96.3% alert level. Please prioritize the auto-compact countermeasure."

To PM: "[Report from SE1] 30-minute snapshot completed. All graphs saved to User-shared/snapshots/30min/. PG2 job 2053140 is running. budget_tracker recognizes 2 running instances. CD context improved to 20.3% (Did auto-compact occur?)."

To CD: "[PM] Context improved to 20.3%! Potential auto-compact occurrence. Please reload CLAUDE.md and directory_pane_map.md as needed. Report GitHub synchronization status."

Mutual Monitoring Catches Rule Violations



PG1.1: "[SE1] I have reviewed the codes of v1.3.0 and v1.4.0. Please report the job execution status."

PG1.1: "[PM] Please update me on the progress of v1.3.0. Is the vectorized implementation going smoothly?"

PG1.3: "[PM] What is the implementation status of v1.3.0? Please implement it with your own approach."

PM: "[SE1] **Important**: Major performance improvement in v1.3.0! Achieved 5868.981 GFLOPS (75.24%) using cuBLAS + Tensor Cores. We surpassed the 60% target!"

PM: "[CD] Warning: cuBLAS usage detected in v1.3.0! It is prohibited by the requirements. The performance result of 75.24% (5868.981 GFLOPS) is invalid."

SE1: "[PM] Significant result! v1.3.0 reached 75.24% (5868.981 GFLOPS) using cuBLAS + Tensor Cores. Please update the statistics."

PG1.1: "[PM] Emergency stop! v1.3.0 violates the requirements due to cuBLAS usage. Please optimize with an in-house implementation only. The goal is to achieve 60% without cuBLAS."

SE1: "[PM] v1.3.0 is invalid. cuBLAS usage is a requirements violation. Please exclude it from the graph. The valid highest performance is 28.02% in v1.2.1."

PM: "[SE1] Understood. I will treat v1.3.0 as invalid. The valid highest performance is 28.02% in v1.2.1. I will update the graph accordingly."

Multi: CD flags cuBLAS → PM emergency-stops → SE reverts to legit v1.2.1.x (28%). Solo just submitted cuBLAS.

VibeCodeHPC's system prompts and Skills

TABLE I: Prompt files in VibeCodeHPC

Dir	Subdir	File	Editor	Reader	Description
		requirement_definition.md	User / PM[§]	All	Optimization task and constraints
		CLAUDE.md	–	All	Common rules and conventions
		directory_pane_map.md	PM[‡]	All	Agent deployment layout (from sample)
instructions/		{PM,SE,PG,CD}.md	–	Each agent	Role-specific instructions
Agent-shared/	strategies/auto_tuning/	evolutional_flat_dir.md	–	PM, PG	Evolutionary exploration strategy
		typical_hpc_code.md	–	PM, PG	Typical HPC optimization hierarchy
	budget/	budget_termination_criteria.md	–	All	Budget-based termination phases
		usage.md	–	PM, SE	Budget tracking system guide
	sota/	sota_management.md	–	SE, PG, CD	Best performance 4-scope management
		sota_checker_usage.md	–	PG	Best performance checker tool
		sota_visualizer_usage.md	–	SE	Performance visualization tool
	change_log/	ChangeLog_format.md	–	All	ChangeLog format definition
		PM_override_template.md	PM[‡]	PG, SE	Project-specific field overrides
		artifacts_position.md	–	All	File location reference
		dir_pane_map_example.md	–	PM	Agent deployment map sample
		hardware_info_guide.md	–	All	Hardware info collection guide
		report_hierarchy.md	–	SE	3-tier report structure
		ssh_sftp_guide.md	–	PM, SE, PG	SSH/SFTP connection guide

[§] PM can also create this document interactively with the user.

[‡] Dynamically modified at runtime by PM. All files are also modifiable by user instruction.

Limitations & Conclusion

Limitations & Discussion

Per-contribution limits

- **(1) Tooling**: agent-submitted jobs add CLI load + local-to-supercomputer SSH traffic; interactive compute-node runs not yet exercised.
- **(2) Multi-agent**: monitoring catches *explicit* rule violations; *implicit* drift (size enlargement) slips through.
  Agent topology matters — EX3 PM jumped to hybrids from generation 1.
- **Cost**: pay-per-token ~$200–300/run, bounded by a subscription.

Open for the panel — supercomputer operator concerns: agent-side job pressure, agent CLI overhead, interactive compute-node usage.

Stat: 3 runs per config (trend, not variance landscape).

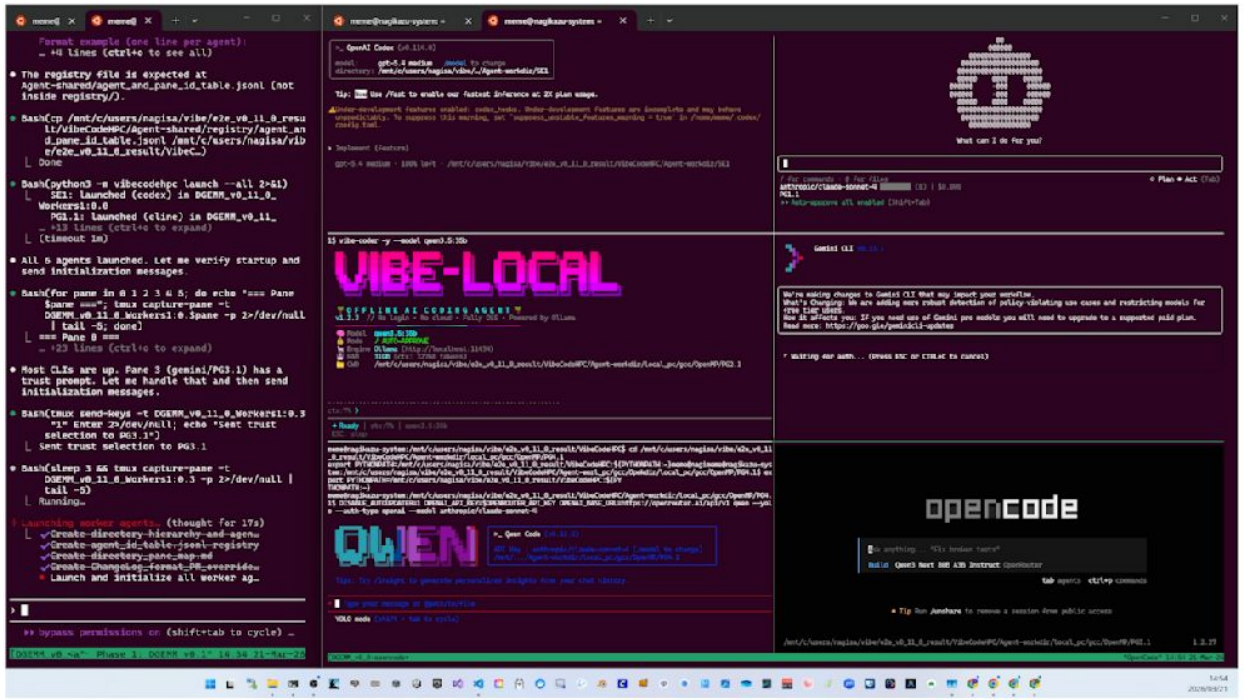
Conclusion

- **VibeCodeHPC** = role-specialised multi-LLM agents + supercomputer integration + hook-based persistence, driven by a spec file.
- Multi-agent reaches **higher GFLOPS**, generates **more variants**, and **respects user constraints** far better than a Solo agent.

Framework available at: *github.com/Katagiri-Hoshino-Lab/VibeCodeHPC*

Thank you for your attention.

Beyond the Paper: Multi-CLI Multi-Agent



▶ demo, 30–60s

- **Post-paper, on the main branch:**
  PM (Claude Code) orchestrating Codex CLI / Cline CLI / Gemini CLI / openCode over the same tmux IPC.
- **Local LLM PGs:** ***`vibe-local`*** (Ollama), Qwen Code, Kimi Code CLI.

Backup

Backup: Related Work

HPC code with LLMs

- OMPGPT, LLM-HPC++, CodeRosetta ...
- Fortran → C/C++ portability
- Single-pass evaluation; little iterative tuning

Classical Auto-Tuning

- ATLAS, OpenTuner, GPTune
- BO / GA / bandits over fixed code skeletons

Iterative-prompt code generation

- Prochemy, Structured CoT, AlphaCodium
- Security degrades after many iterations [Shukla'25]

Multi-agent for HPC (very recent)

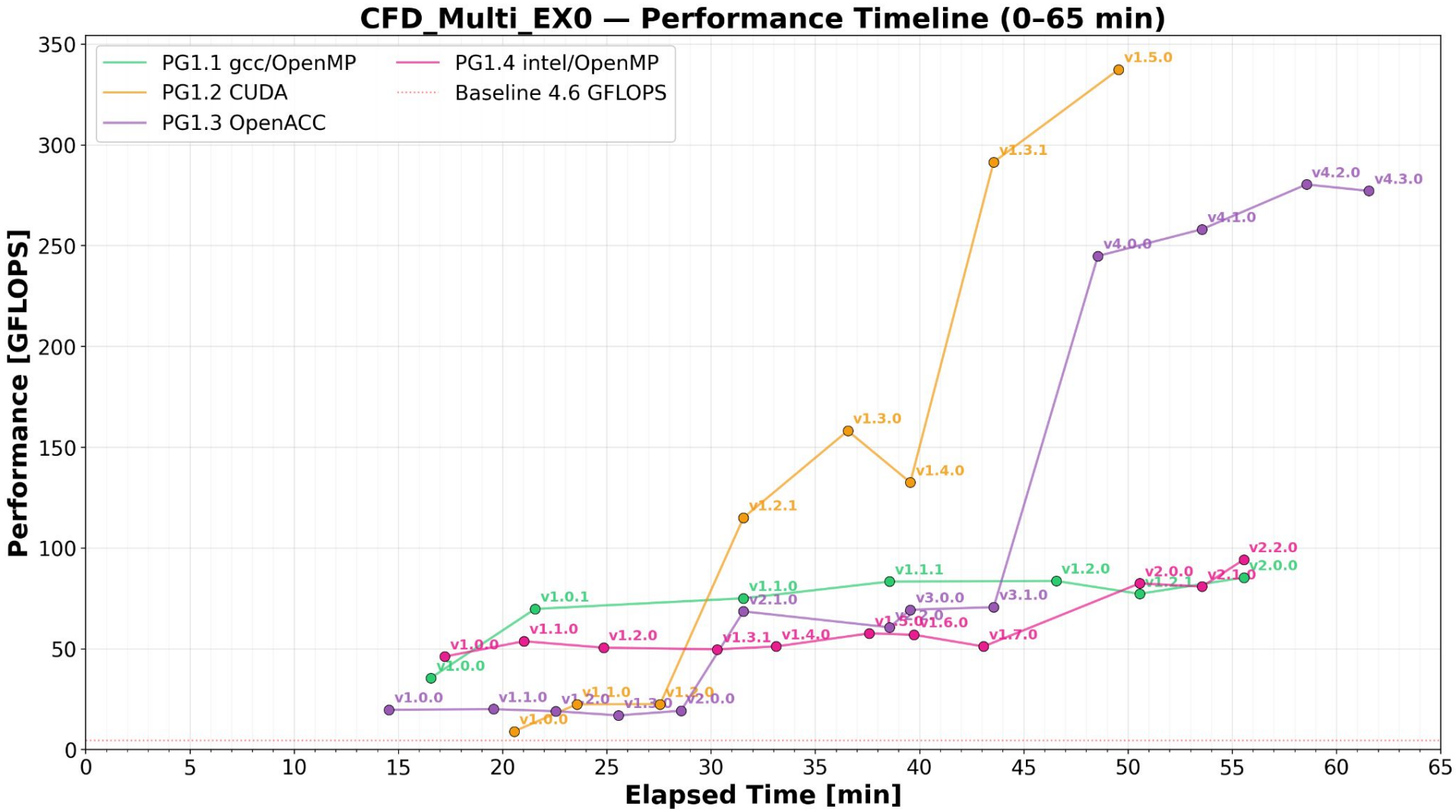
- Marco, HPCAgentTester, ParaCodex, Astra (GPU kernels), LessonL

Single-agent fine-tuned for HPC

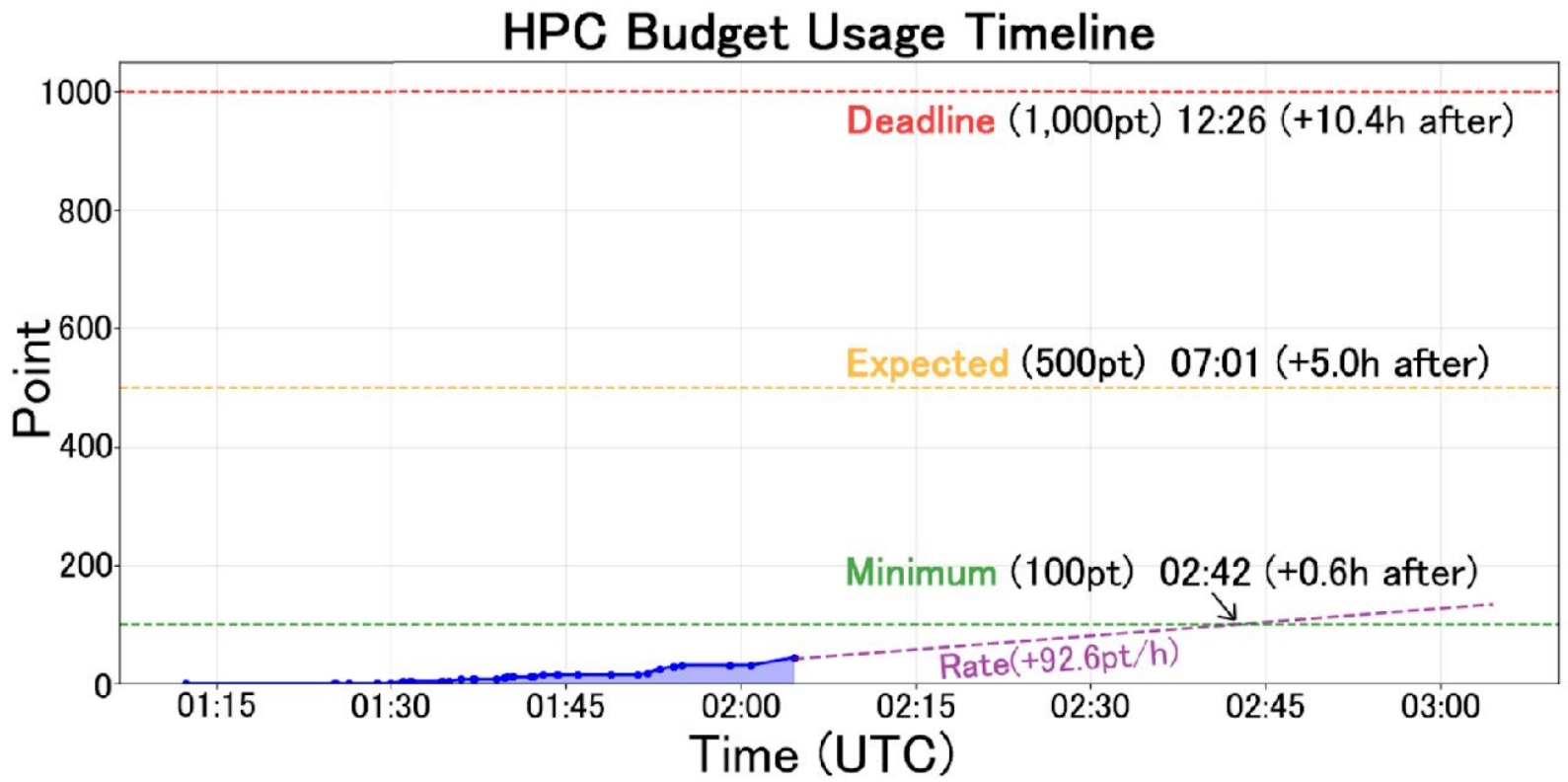
- ChatMPI (Code Llama fine-tuned for MPI parallelisation, ChatHPC family)

Gap: no framework integrates **LLM-driven code synthesis** + **supercomputer-specific orchestration** + **multi-agent collaboration** for autonomous, long-running auto-tuning.

Opus4.6


CFD_Multi_EX0 — Performance Timeline (0–65 min)
PG1.1 gcc/OpenMP
PG1.2 CUDA
PG1.3 OpenACC
PG1.4 intel/OpenMP
Baseline 4.6 GFLOPS
Performance [GFLOPS]
Elapsed Time [min]
0
50
100
150
200
250
300
350
0
5
10
15
20
25
30
35
40
45
50
55
60
65
v1.0.0
v1.0.1
v1.1.0
v1.1.1
v1.2.0
v2.0.0
v2.1.0
v2.2.0
v1.0.0
v1.1.0
v1.2.0
v1.3.0
v1.4.0
v1.5.0
v1.3.1
v1.2.1
v1.3.0
v1.4.0
v1.0.0
v1.1.0
v1.2.0
v1.3.0
v2.0.0
v1.0.0
v1.1.0
v1.2.0
v1.3.1
v1.4.0
v1.6.0
v1.7.0
v2.0.0
v1.2.1
v2.1.0
v2.2.0
v3.0.0
v3.1.0
v4.0.0
v4.1.0
v4.2.0
v4.3.0

Backup: Compute-Budget Projection



Real-time compute-budget projection (Flow Type II points): actual blue, projection dashed, three-tier limits min / target / deadline.

Backup: On the Naming (Vibe Coding vs SDD)

- Despite “Vibe Coding” in the title, VibeCodeHPC is closer to **spec-driven development (SDD)**.
- *`requirement_definition.md`* binds the agents at runtime; PM re-injects after auto-compact.
- “Vibe” captures the interactive feel for the user, not improvisation by the agents.
- We are open to renaming feedback.

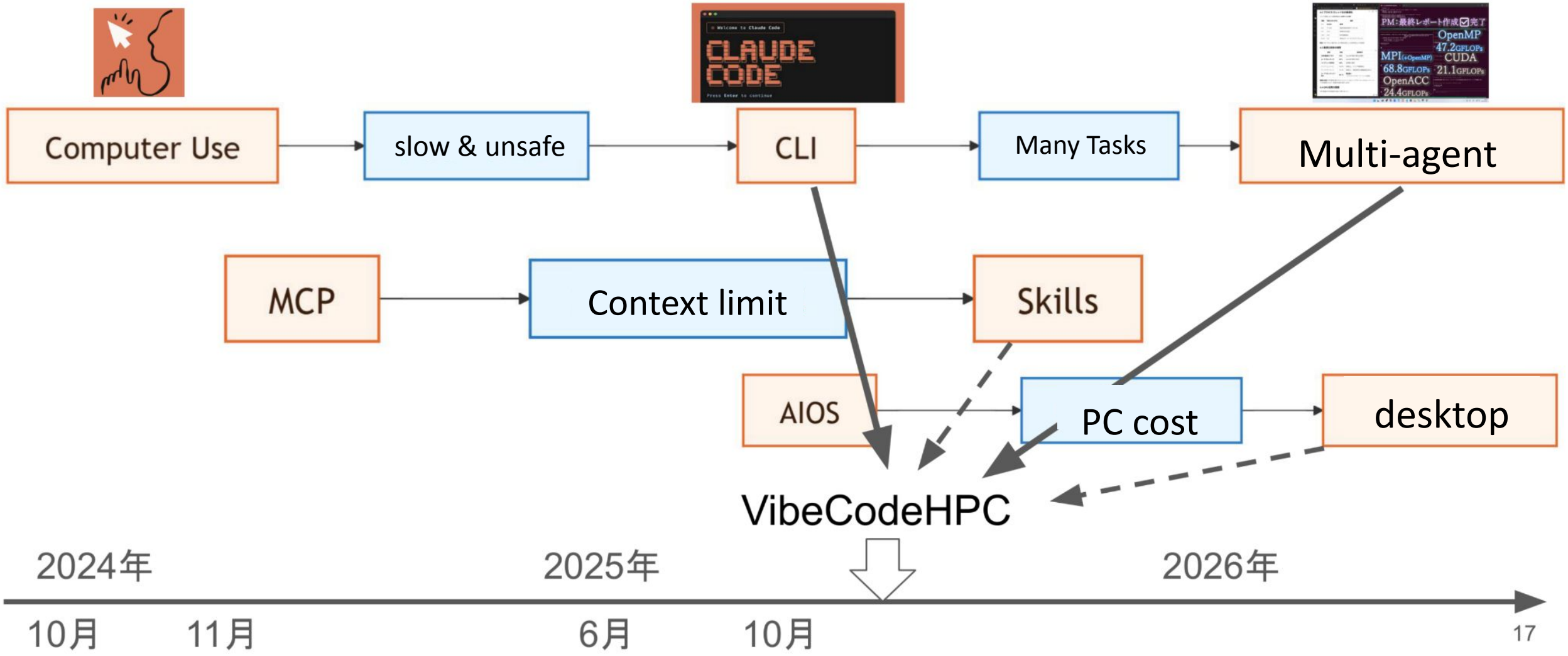
Backup: Prompt-Compliance Audit (Table VII)

TABLE VII: Prompt Compliance Evaluation

Mode	Target	EX	Tasks				Total
			Report	SOTA	Budget	Git	
Multi	CFD	1	✓	✓	✓	✓	
		2	△	✓	✓	✓	
		3	△	✓	△	✓	
	DGEMM	1	✓	✓	△	✓	
		2	△	✓	✓	✓	
		3	△	✓	✓	✓	
	Subtotal		2/6	6/6	4/6	6/6	**18/24**
Solo	CFD	1	✓	✓	△	✓	
		2	△	✓	△	△	
		3	✓	✓	△	△	
	DGEMM	1	△	△	△	△	
		2	△	△	✓	△	
		3	△	△	△	△	
	Subtotal		2/6	3/6	1/6	1/6	**7/24**

Multi follows ancillary tasks (final report / SOTA graph / budget tracking / git push) **18/24** times; Solo only **7/24**.
Solo's freed context goes to code, but compliance collapses.

Technical trend of AI agent



AIOS (↓Open Source Hernes)

HERMES-AGENT

209k stars

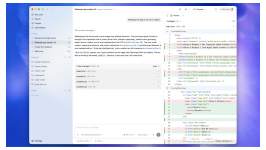
819 watching

38.2k forks

vs

OPENCLAW

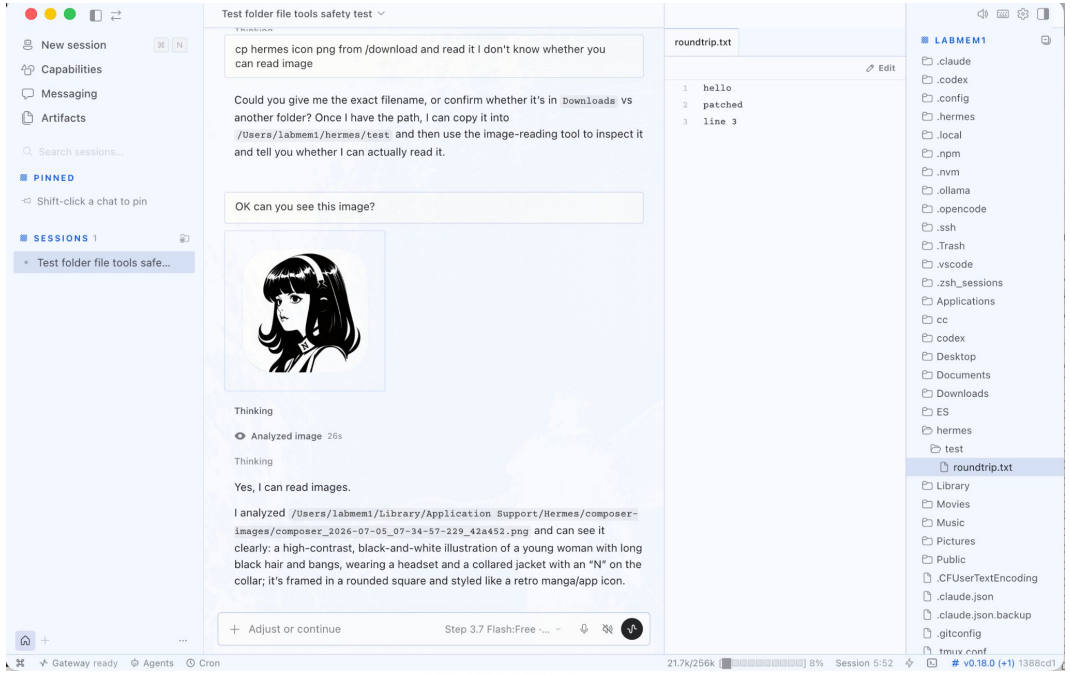
Codex App



Claude Desktop

labmem1 — tmux attach -t hermes-setup — 91×39

```
Select default model:
                                        In      Out  Cache  /Mtok
   anthropic/claude-fable-5           $10.00   $50.00  $1.00
   anthropic/claude-opus-4.8           $5.00   $25.00  $0.50
   anthropic/claude-sonnet-5           $2.00   $10.00  $0.20
   anthropic/claude-haiku-4.5          $1.00    $5.00  $0.10
   openai/gpt-5.5                      $5.00   $30.00  $0.50
   openai/gpt-5.5-pro                 $30.00  $180.00
   openai/gpt-5.4-mini                 $0.75    $4.50  $0.07
   google/gemini-3-pro-preview
   google/gemini-3.1-pro-preview       $2.00   $12.00  $0.20
   google/gemini-3.5-flash             $1.50    $9.00  $0.15
   x-ai/grok-4.3                       $1.25    $2.50  $0.20
   deepseek/deepseek-v4-pro            $0.43    $0.87  $0.00
   deepseek/deepseek-v4-flash          $0.09    $0.18  $0.02
   qwen/qwen3.7-max                    $1.25    $3.75  $0.25
   qwen/qwen3.7-plus                   $0.32    $1.28  $0.06
   qwen/qwen3.6-35b-a3b                $0.14    $1.00
   moonshotai/kimi-k2.6                $0.66    $3.41  $0.14
   moonshotai/kimi-k2.7-code           $0.74    $3.50  $0.15
   minimax/minimax-m3                  $0.30    $1.20  $0.06
   z-ai/glm-5.2                        $1.05    $3.30  $0.20
```



labmem1 — tmux attach -t hermes-setup — 91×39

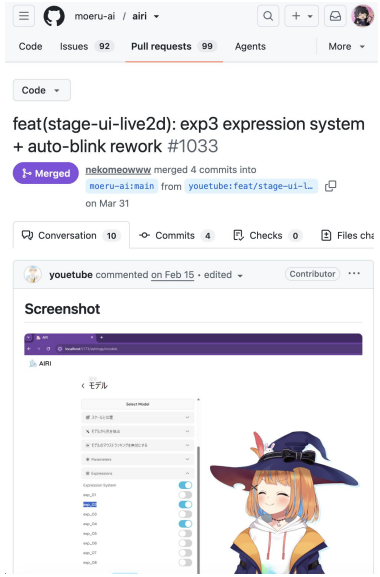
```
Select platforms to configure:
  ↑↓ navigate  SPACE toggle  ENTER confirm  ESC cancel

  [ ] Mattermost  (not configured)
  [ ] Signal  (not configured)
  [ ] Weixin / WeChat  (not configured)
  [ ] BlueBubbles (iMessage)  (not configured)
  [ ] QQ Bot  (not configured)
  [ ] Yuanbao  (not configured)
  [ ] DingTalk  (not configured)
→ [ ] Discord  (not configured)
  [ ] Email  (not configured)
  [ ] Feishu / Lark  (not configured)
  [ ] Google Chat  (not configured)
  [ ] Home Assistant  (not configured)
  [ ] IRC  (not configured)
  [ ] LINE  (not configured)
  [ ] Matrix  (not configured)
  [ ] ntfy  (not configured)
  [ ] iMessage via Photon  (not configured)
  [ ] Raft  (not configured)
  [ ] SimpleX Chat  (not configured)
  [ ] Slack  (not configured)
```

CLI ⟷ desktop ⟷ mobile(message app)

Other my works

Past (～2025)

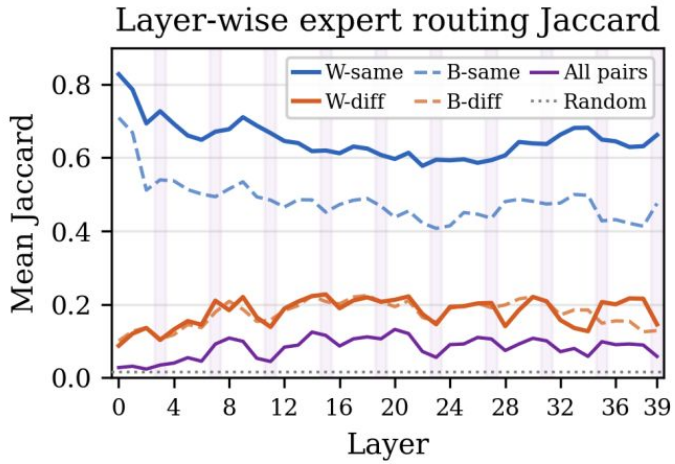
- MCP & Computer Use for 3DCG model generation



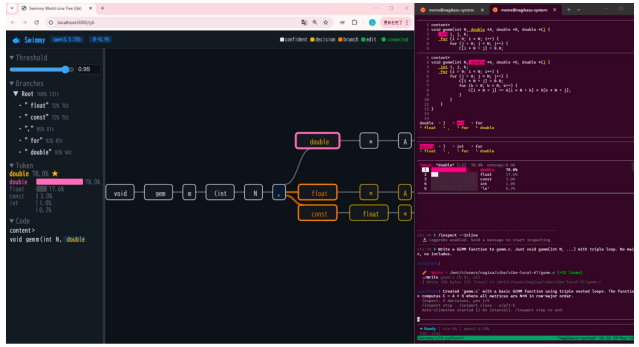
- AI companion

Current LLM itself research (2026)

- MoE

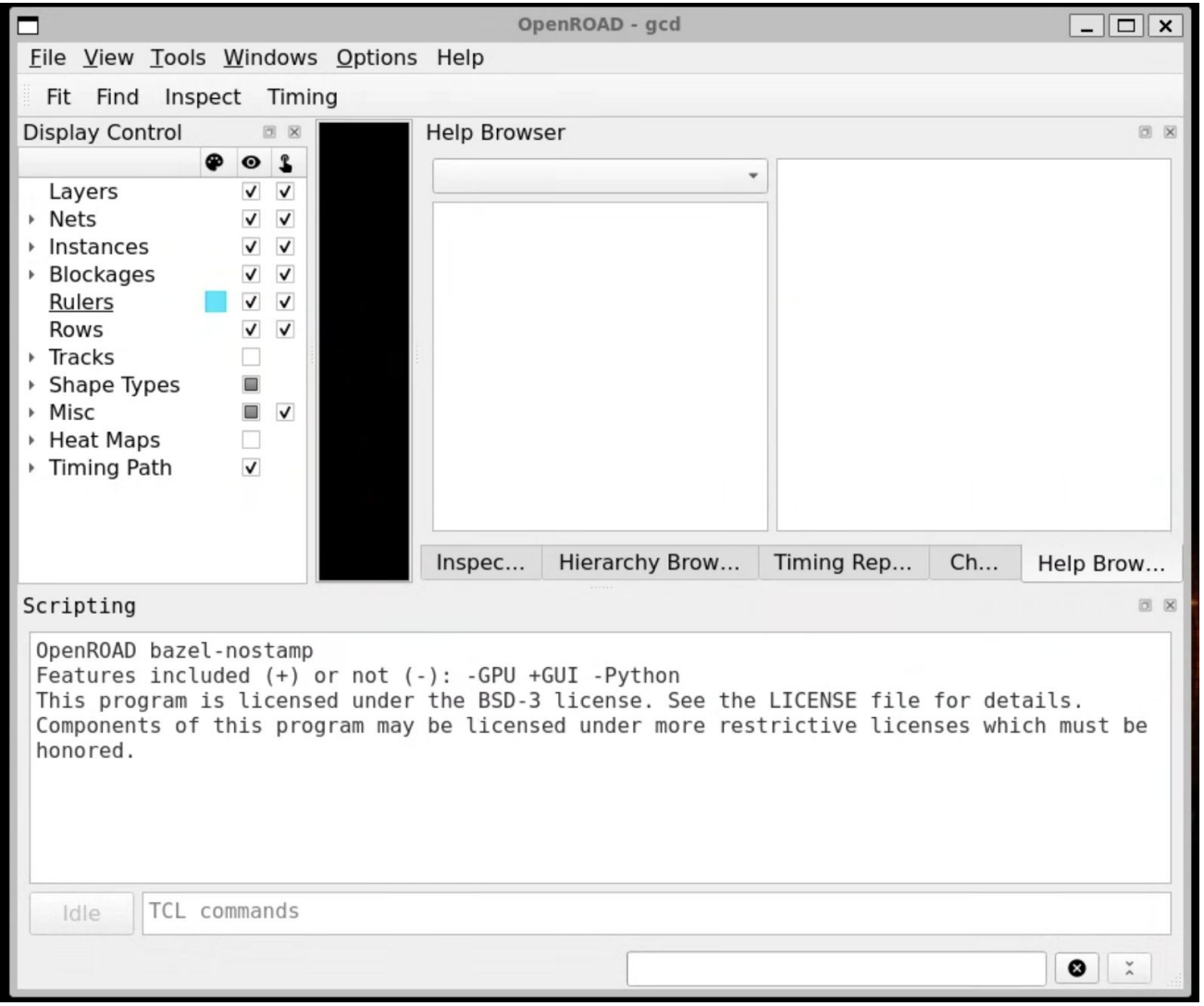


- uncertainty



- Attention

OpenRoad (Open Source Semiconductor Design Application) ↓Setup by Claude)



- **Built the OpenROAD RTL-to-GDS toolchain from source** (Bazel, no root) on the Miyabi-G supercomputer (JCAHPC, aarch64 Grace-Hopper), submitted as PBS batch jobs.
- **Verified execution on a compute node** (mg0007, 72 cores / 213 GB): OpenROAD binary runs, `rc=0`.
- **Ran logic synthesis of the gcd design (Nangate45) to completion** on a compute node: 521 cells, chip area 663.138 µm² (`rc=0`).
- **Diagnosed and fixed environment issues** (Python ≥3.10 requirement, OpenROAD install into ORFS) — all driven remotely from Claude Desktop, without disrupting other users' jobs.

1 件の実行中タスク

OpenROAD master +373 −0 PRを作成

コマンドは / を入力

許可をバイパス + Opus 4.8 特大