VibeCodeHPC: A Multi-LLM Agent Auto-Tuner for HPC Codes

Shun-ichiro Hayashi[1], Koki Morita[1], Daichi Mukunoki[2], Tetsuya Hoshino[2], Takahiro Katagiri[2]
[1]Graduate School of Informatics, Nagoya University, Japan
[2]Information Technology Center, Nagoya University, Japan

1. Background：Challenges of Applying CLI-Based LLM Agents to HPC Workflows

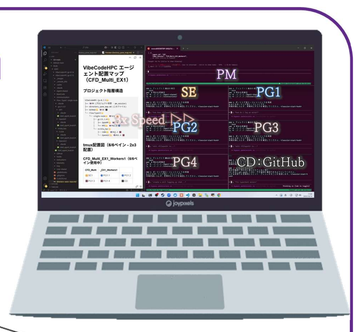
While CLI-based LLM agents like **Claude Code** enable automated environment setup, code editing, and debugging, their direct application to **supercomputers** remains difficult due to system-specific **constraints** such as complex **SSH** authentication, **vendor-specific** commands, restricted software installation, and **budget**-aware execution.

HPC program development is a highly complex **exploratory** task that requires simultaneous consideration of **performance**, distributed **heterogeneous** hardware architecture, and numerical **correctness**.

To address these challenges, we propose VibeCodeHPC[1][2], a CLI-based **multi-agent framework** that enables iterative prompting, mutual inspection, and **auto-tuning** of arbitrary code across heterogeneous HPC environments.

2. Dynamic Agent Organization

Each **Claude Code** agent communicates with others via shell script wrapping **Terminal Multiplexer** (**tmux**) functions



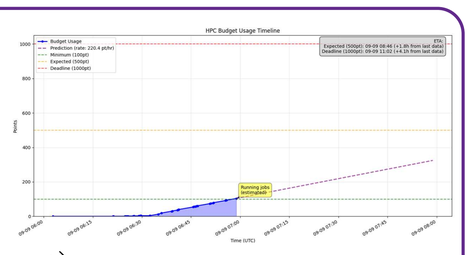
3. Project Structure

User prepares **3 items**

```
VibeCodeHPC/ PM
├── requirement_definition.md  # User prompt
├── _remote_info/              # HPC info
├── BaseCode/                  # existing code
├── User-shared      # reports and graphs
├── Agent-shared     # prompts
├── CLAUDE.md        # Common Prompt
├── GitHub/ CD       # Anonymized copy
└── Flow/TypeII/single-node/ SE1
    ├── hardware_info.md  # size, networks…
    ├── intel/
    │   ├── OpenMP/ PG1.1
    │   └── MPI/ PG1.2
    ├── gcc/
    └── CUDA/ PG1.3
        ├── ChangeLog.md
        └── ssh_sftp_session.json
```

Spawn by **PM**

Other implemented features



☑ **Hooks** prevent LLM from sleeping
☑ Manage supercomputer's **Budget**

Roadmaps

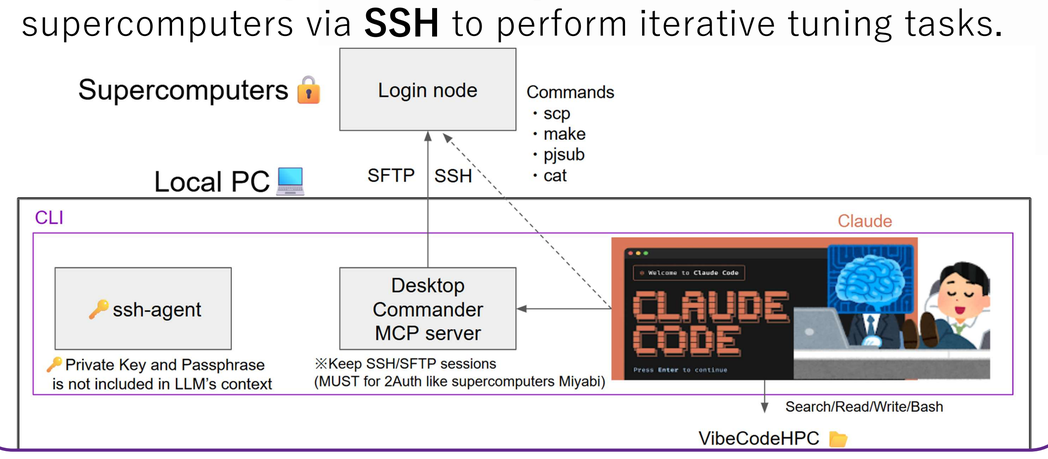
☐ Refactor as Skills （Multi CLIs support）
☐ Local LLM inference and training

URLs (arXiv, GitHub)

[1] Shun-ichiro Hayashi, Koki Morita, Daichi Mukunoki, Tetsuya Hoshino, Takahiro Katagiri, "**VibeCodeHPC**: An Agent-Based Iterative Prompting Auto-Tuner for HPC Code Generation Using LLMs", arXiv: 2510.00031, 2025.
[2] Shun-ichiro Hayashi, VibeCodeHPC, https://github.com/**Katagiri-Hoshino-Lab/VibeCodeHPC-jp**

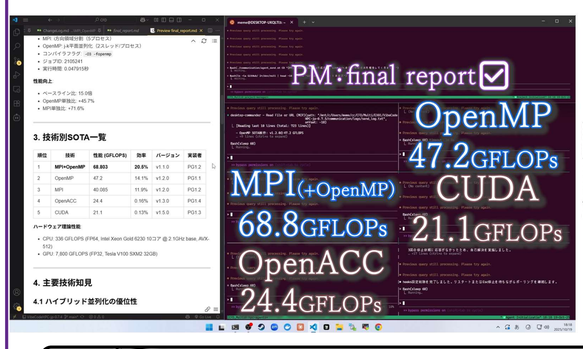
4. System Configuration

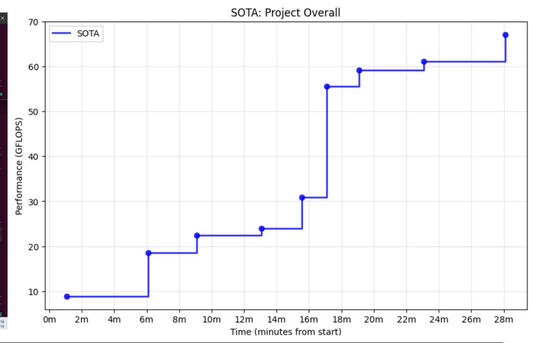
VibeCodeHPC runs on a **local PC** and launches multiple **Claude Code** agents. These agents interact with supercomputers via **SSH** to perform iterative tuning tasks.

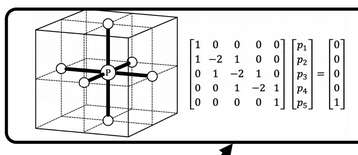


5. Experiments (2 targets x3 runs)

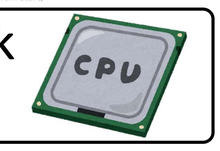
We compared **Multi**-agents & **Solo** agent to evaluate robustness in complex exploratory HPC tasks.







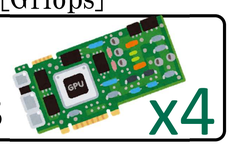
Himeno Benchmark
a 10-core CPU



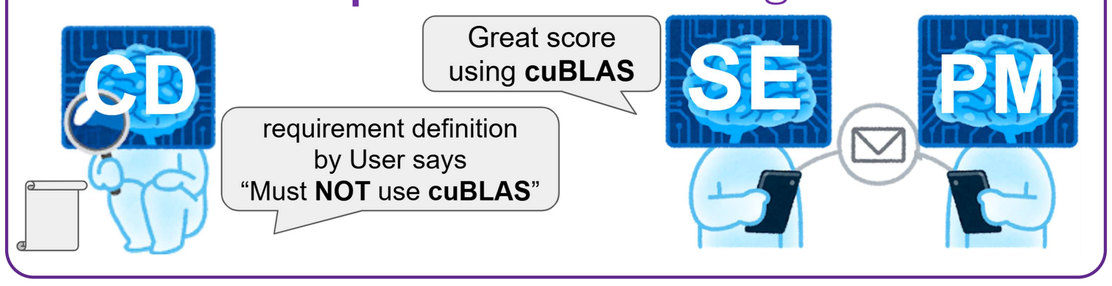
Target	Mode	EX1	EX2	EX3	Avg
CFD	Multi	68.8	77.8	67.1	**71.23**
	Solo	21.5	23.9	39.9	28.43
DGEMM	Multi	1385	3677	916	**1993**
	Solo	805	607	1098	837

[Gflops]

(A)(B)=(C) Matrix product
NVIDIA V100 GPUs x4



6. mutual **inspection** of Multi-agents



7. mutual **cooperation** of Multi-agents

Multi-agents vs **Solo agent**
(Mutual care after auto-compact) (Easily forget tasks)

SCA/HPCAsia2026 Poster session Stage 1, Jan, 26~29, at Osaka

Acknowledgements: This work was supported by the Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures (JHPCN) and the High Performance Computing Infrastructure (HPCI) under Project ID jh250015. It was also partially supported by JSPS KAKENHI Grant Numbers JP23K11126 and JP24K02945. In addition, this work was supported by the JST Research and Development Program for Next-generation Edge AI Semiconductors (Grant Number JPMJES2511).