

HPC-GENIE Project: Towards Code Generative AI for HPC Programming

Takahiro Katagiri (Information Technology Center, Nagoya University,
E-mail: katagiri@cc.nagoya-u.ac.jp)

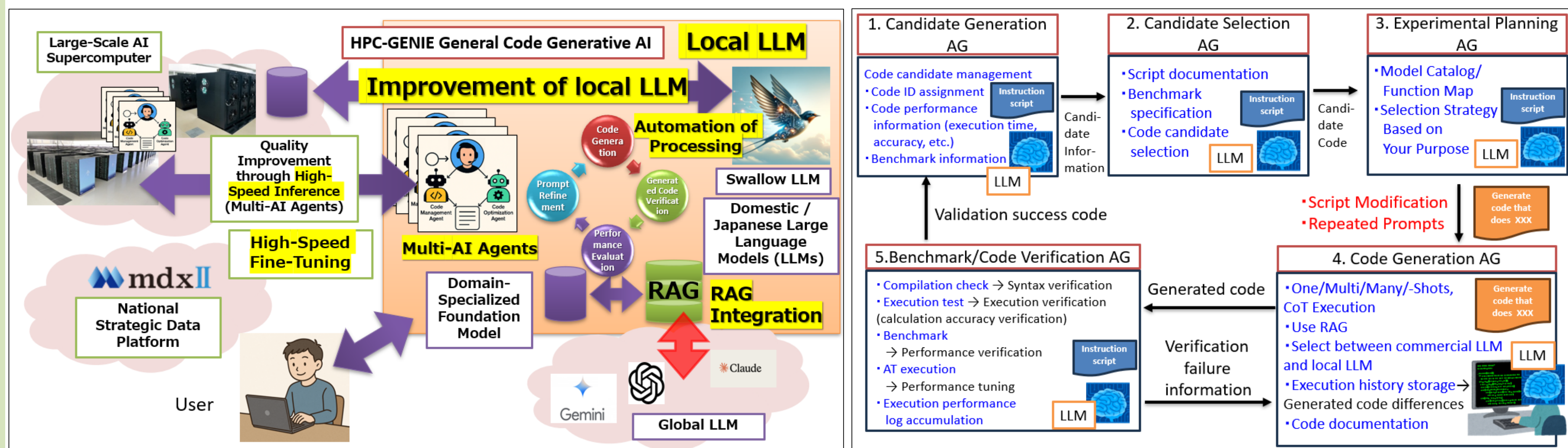
Aim of the Project

- The HPC-GENIE (High-Performance Computing with Generative Neural Intelligence for Execution) project is an initiative launched by members of Information Technology Center and Graduate School of Informatics at Nagoya University.
- It is a research project that leverages **code-generating AI** for the **automatic generation of HPC programs**. By integrating **prompt engineering** based on large language models (LLMs) **with software auto-tuning (AT)** technologies, the project aims to achieve automation that **dramatically enhances the productivity of HPC software development**.



Automation of Local LLM-Specialized Processing by HPC-GENIE

- **Automation of Code Tuning Process**: iterative prompting of code generation → generated code verification → performance evaluation → prompt refinement
- Enhancement of code quality through multi-AI-agent (A2A) collaboration
- **RAG-based** reinforcement of local LLMs
- Improvement of generated code quality via **supercomputer-accelerated fine-tuning and inference** (**multi-AI-agent execution**) for local LLMs



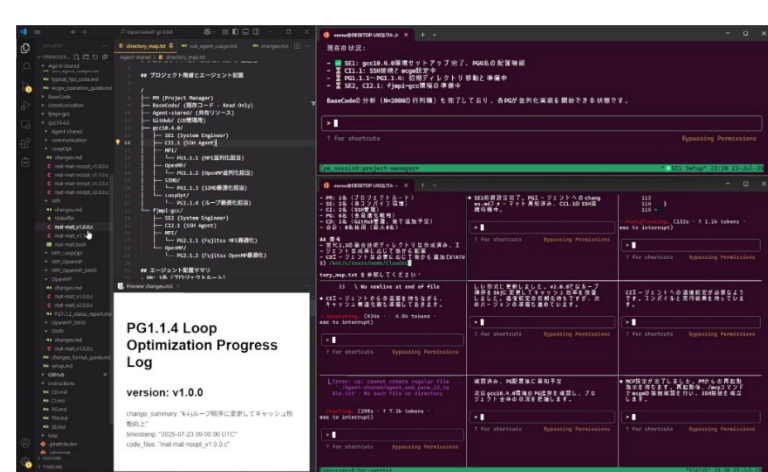
HPC-GENIE: A General Code Generative AI System

Iterative Promompt for the HPC-GENIE System

Overview of VibeCodeHPC

- **VibeCodeHPC: Multi-Agent System for Auto-Tuning of HPC Code Optimization**

- **A Case Study: Multi-AI agents code opination for matrix-matrix multiplication**



Launch screen

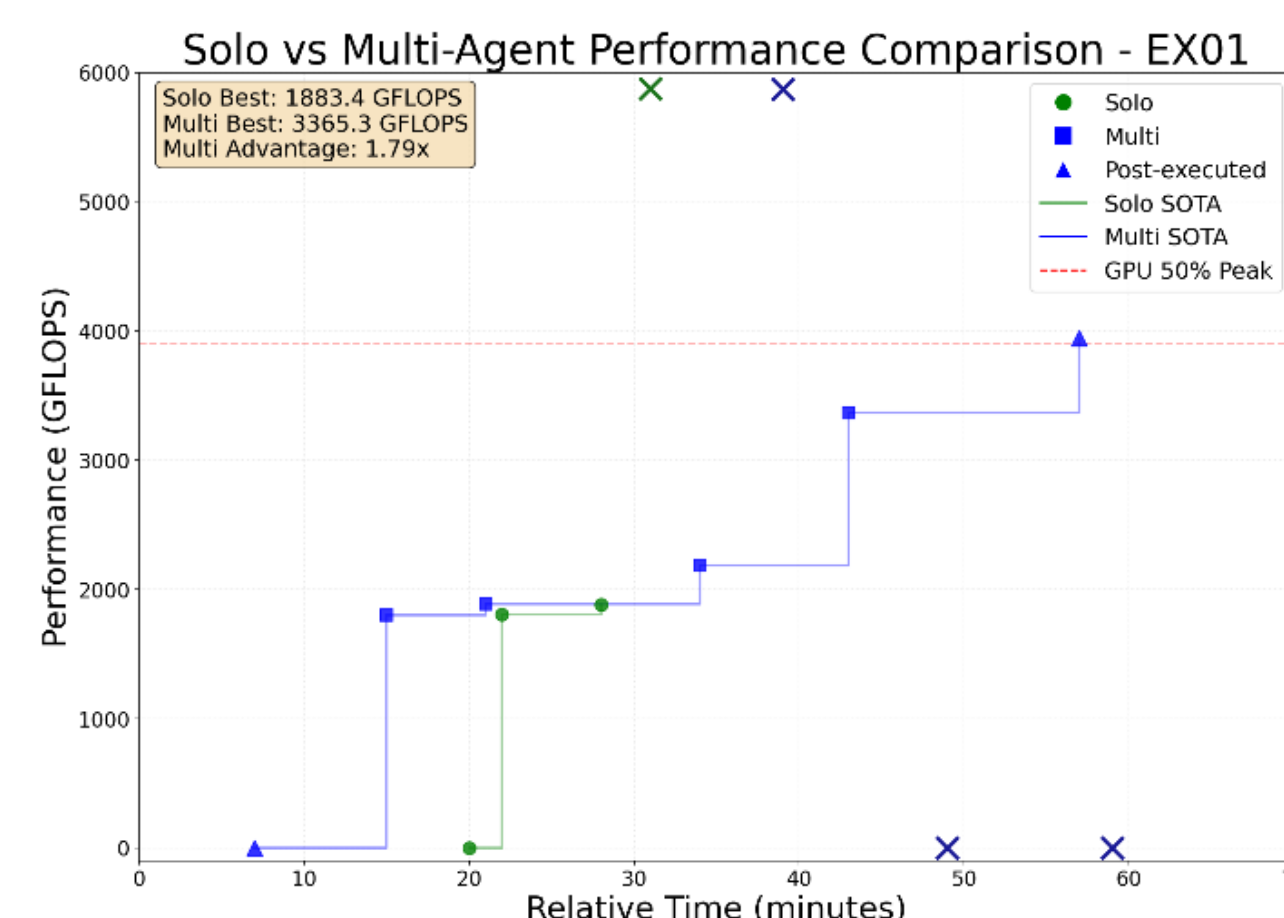
Roles of Agents

Agents	Roles	Scope of Responsibility
PM	Project Management	Requirements Definition, Resource Allocation, and Budget Management
SE	System Design	Agent Monitoring, Statistical Analysis, and Report Generation
PG	Code Generation and Execution	Parallel Implementation, SSH/SFTP Connection, Job Execution, Performance Measurement, and SOTA Evaluation
CD	Deployment Management	Publication and Anonymization of SOTA-Achieving Code

Original Code
(Before Optimization)

```
void gemm_naive (int M, int N, int K,
double alpha, const double* A, int lda,
const double* B, int ldb,
double beta, double* C, int ldc) {
for (int i = 0; i < M; i++) {
for (int j = 0; j < N; j++) {
double sum = 0.0;
for (int k = 0; k < K; k++) {
sum += A[i * lda + k] * B[k * ldb + j];
}
C[i * ldc + j] = alpha * sum + beta * C[i * ldc + j];
}
}
```

Optimizaztion History



A multi-agent system generates high-performance code faster than a solo-agent system and additionally achieves even higher performance.

VibeCodeHPC:

Source Code: <https://github.com/Katagiri-Hoshino-Lab/VibeCodeHPC-jp>
arXiv: <https://www.arxiv.org/abs/2510.00031>

Information Technology Center, Nagoya University
<http://www.icts.nagoya-u.ac.jp/en/center/>