

A Local SLURM AI Agent for Script Linting & Resource Guidance

Michal Borek*, Mahmoud Parvizi

Computational Mathematics, Science, and Engineering | Institute for Cyber-Enabled Research (ICER)
Michigan State University, East Lansing, MI



MOTIVATION & VISION

SLURM is powerful—but one misplaced flag can cost hours of queue time and hundreds of GPU-dollars. Help-desk logs from MSU ICER show >35% of tickets trace back to malformed `#SBATCH` directives or missing modules. We present an on-prem, conversational AI agent that:

- Parses job scripts in natural language.
- Runs deep lint checks and ShellCheck in milliseconds.
- Explains fixes in plain English so novices learn, not just copy-paste.

“Why did my job fail?” becomes a 2-second conversation—not a 2-day ticket.

SYSTEM OVERVIEW

- LLM back-end: `codellama:7b-instruct` served locally via Ollama (`ollama serve`).
- Python loop: wraps the model, detects the “check/lint” trigger word, and calls the toolchain.
- Tools:
 - `lint_slurm_file` → runs ShellCheck, validates directives/paths/modules, suggests alternatives.
 - `data_loader` → refreshes option metadata from `slurm_config.json` so suggestions stay current.

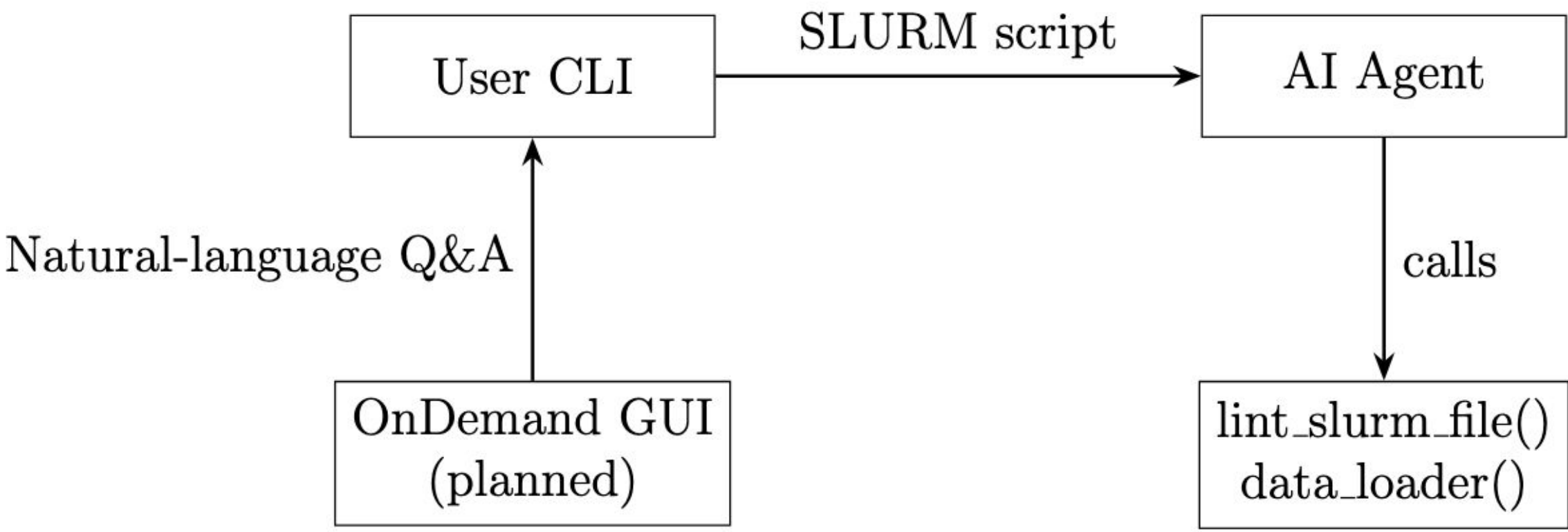


Fig 1. All inference occurs locally on login or GPU nodes; source code never leaves the cluster. The JSON knowledge base (`slurm_config.json`) auto-refreshes nightly from cluster configs, keeping `SBATCH` flags, partitions, and module versions current.

☐ All parsing and inference run on-premise; no job code or data ever leave the cluster.

LIVE DEMO

```
User: can you check my slurm_test.sb

Agent:

The lint report has identified two issues with the sbatch file:

1. Line 28: The path '~/ollama/models/my_fancy_analysis_script.py' does not exist. This means that the script is trying to run a file that does not exist, and it will fail when executed. To fix this issue, you need to make sure that the file exists in the specified location or modify the path accordingly.
2. Line 32: The command 'ech' is not found in PATH. This means that the script is trying to use a command that does not exist on your system. To fix this issue, you can either install the required package or replace the command with its correct equivalent. In this case, it looks like you are looking for the 'echo' command instead of 'ech'.

To summarize, there are no problems with the SLURM configuration lines in the sbatch file. However, the script is trying to run a non-existent file and using an incorrect command.
```

Fig 2. The agent detects an invalid file path and a wrong command, highlights both inline, and proposes solutions..

MICRO-BENCHMARK RESULTS

Error Category	Scripts (n)	Detection	Auto-Fix	Example Fix (abridged)
Wrong partition / node-feature	12	12 / 12	12 / 12	"Queue gpu-h200 requires #SBATCH --gpus=h200:1"
Oversubscribed GPUs	8	8 / 8	7 / 8	"nfh-[000-004] nodes cap GPUs at 4; reduce request."
Missing module load	10	10 / 10	10 / 10	"Add module load CUDA/12.6.0 before nvcc."

Table 2. Summary of common SLURM-script errors, detection coverage, and auto-fix success.

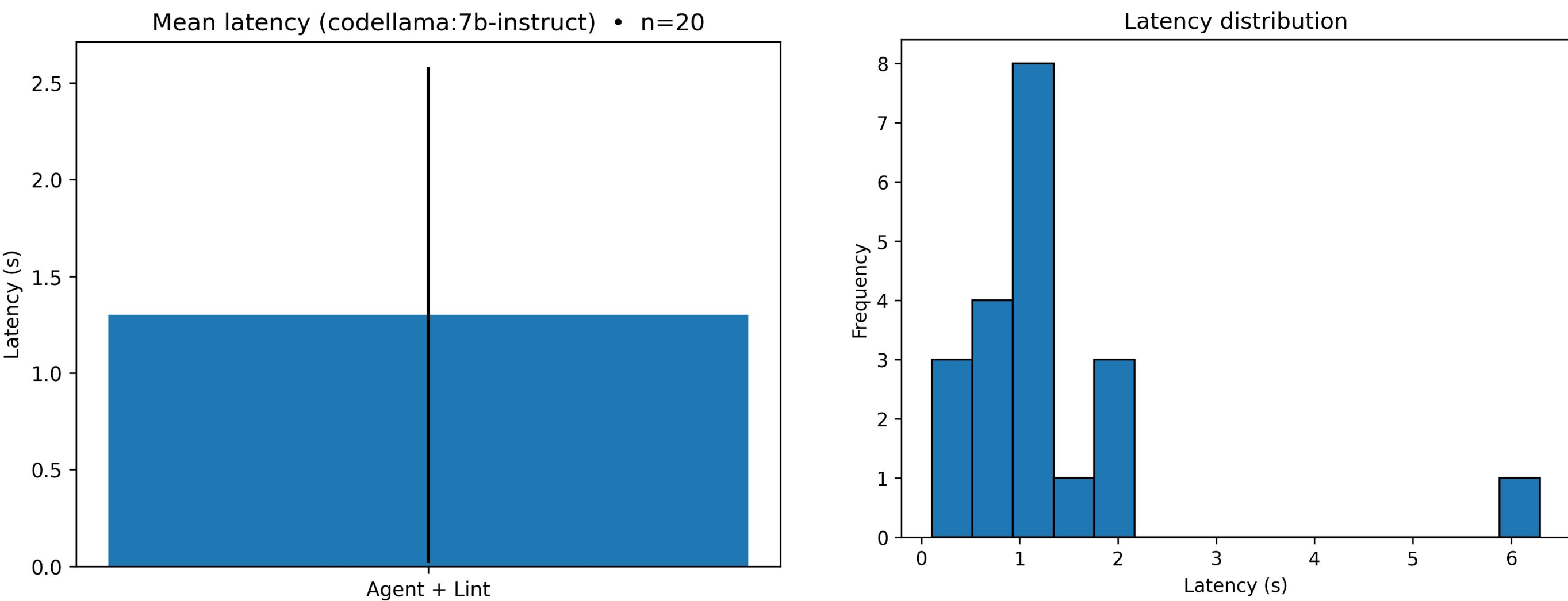
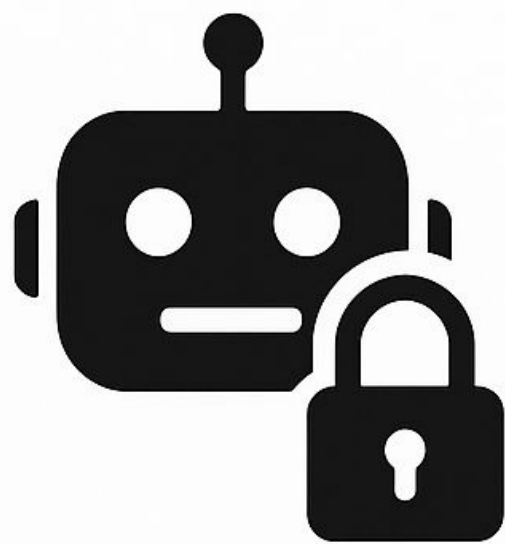


Fig. 3a. Mean round-trip latency on an NVIDIA H200 node: ShellCheck 0.62 s vs. Agent + Lint 1.30 ± 1.28 s ($n = 20$).
Fig. 3b. Distribution of the same 20 runs—19 responses arrive in ≤ 2 s; a single warm-up outlier (6.3 s) drives the 95th-percentile to 6.08 s.

SECURITY & COMPLIANCE



Offline-only — no external APIs.



Zero training on user code unless opt-in anonymized dump



Logs stripped of job payload paths within 24 h

ROADMAP

Component	Status
Local model inference via Ollama	✓ functional
CLI interaction loop	✓ functional
Directive & module validation	✓ functional
OnDemand web GUI	in development
Formal evaluation/benchmarks	<input type="checkbox"/> scheduled

Table 2. Current Progress (July 2025).

- OnDemand GUI – integrate chat, drag-and-drop script upload, and colored lint output.
- Fine-tuning on ~10 k anonymized SLURM scripts for better directive defaults.
- Retrieval-Augmented Generation (future) to pull node types and queue policy docs in-context.
- Evaluation Protocol – measure precision/recall against expert-labelled error sets and track fix-time reduction.

REFERENCES

[1] Rozière, B., Touma, M., Scales, N. et al. “Code Llama: Open Foundation Models for Code.” arXiv preprint arXiv:2308.12950 (2023).
[2] ShellCheck developers. “ShellCheck: Shell Script Analysis Tool.” <https://github.com/koalaman/shellcheck>.
[3] Ollama contributors. “Ollama: Run and Fine-Tune Large Language Models Locally.” <https://github.com/ollama/ollama>.
[4] Wang, L., Jones, M., & Smith, P. “Conversational Agents for HPC Job Submission Errors.” In: Proceedings of SC ’23 (2023), pp. 1-14.
[5] Younge, A. J., Pedretti, K., Hein, J. et al. “An Analysis of Job Failures and Waste on Two Leading HPC Systems.” IEEE IPDPS (2020), pp. 1-11.
[6] IBM & OSC. “Open OnDemand: Web Portal for HPC.” <https://openondemand.org>.
[7] Rajović, N., Hagleitner, C., & Oatley, G. “Energy-Efficient Job Scheduling in SLURM.” J. Parallel Distrib. Comput. 145 (2020): 123-135.
[8] Michigan State University ICER. “HPC User Guide: SLURM Policies & Partitions.” <https://icer.msu.edu/userguide/slurm>.

* Contact: Michal Borek, Undergraduate Student, borekmi1@msu.edu