Towards AI-Driven Interfaces for Scientific Data Management

Jaime Cernuda
Shazzadul Islam
jcernudagarcia@hawk.illinoistech.edu
sislam6@hawk.illinoistech.edu
Illinois Tech
Chicago, Illinois, USA

Isa Muradli
Luke Logan
imuradli@hawk.illinoistech.edu
llogan@hawk.illinoistech.edu
Illinois Tech
Chicago, Illinois, USA

Xian-He Sun
Anthony Kougkas
sun@illinoistech.edu
akougkas@illinoistech.edu
Illinois Tech
Chicago, Illinois, USA

1 Extended Abstract

Exploring and producing scientific data requires researchers to navigate a complex ecosystem of disconnected tools across data gathering, resource allocation, application deployment, and data analysis. Large language models (LLMs) promise to simplify HPC tasks through natural language interfaces. Instead of writing an `sbatch` script with multiple flags, a scientist can prompt: "Run my simulation on 32 cores for three hours". While researchers have used models to perform HPC actions like analyzing HDF5 outputs [4] or generating Slurm scripts [2], these methods depend on LLMs having been trained on the target system, causing reliability and portability issues.

Recent advancements have introduced agentic systems, where a software orchestrator enables continuous prompting and multiple inferences per query. Within these agentic frameworks, LLMs can request a call to a tool, predetermined code executed by the orchestrator whose results are placed into the model's context. The Model Context Protocol (MCP) [1] was introduced as a standard to define and interact with these tools across different models. Although cloud and general-purpose ecosystems now offer dozens of MCPs, scientific computing remains underrepresented with only isolated tools visualization [3, 5].

To address this issue, we introduce *IOWarp-mcps*, a suite of MCPs designed to enable AI-driven scientific data management. IOWarp-mcps supports all stages of the scientific data management workflow from stage-in, resource allocation, application execution, interfaces to scientific data formats, or visualization. IOWarp-MCPs employ two core design principles to adapt to the requirements of large-scale scientific datasets. First, *chunked I/O access* mitigates memory constraints by partitioning variables along their primary dimension into dynamically-sized segments according to the model capabilities, tracking metadata to enable lazy evaluation, and preventing memory exhaustion. Second, *label-based filtering* applies selective data reduction at the tool-output interface before AI model ingestion, significantly reducing context overhead while preserving the needed semantics.

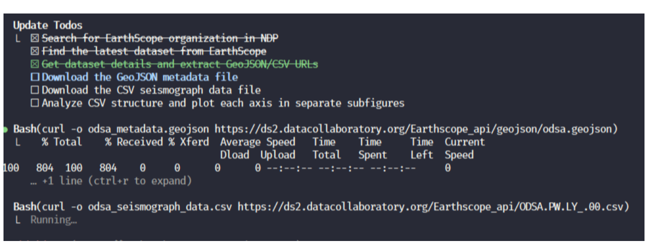

Figure 1: NDP-MCP: Automated discovery and stage-in of EarthScope seismic datasets from the National Data Platform.

We evaluated IOWarp-MCP across data management scenarios.

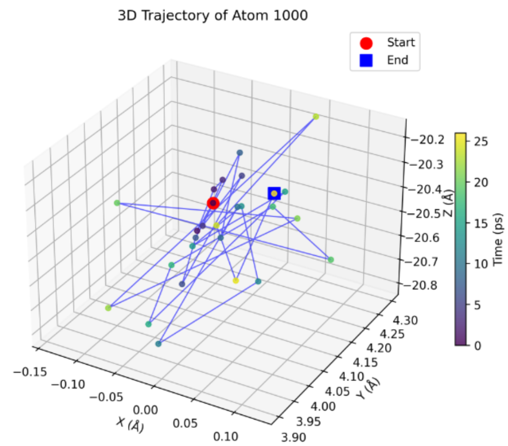

Figure 2: ADIOS-MCP: Exploration and trajectory analysis of LAMMPS BP5 molecular dynamics data.

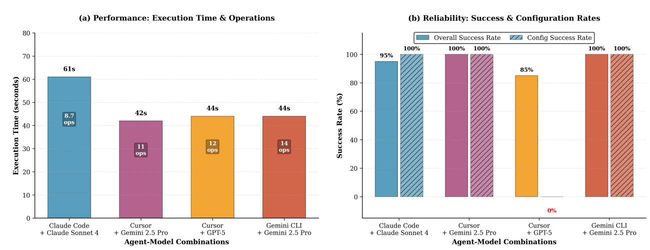

Figure 3: Jarvis-MCP: Automated deployment and configuration of IOR parallel I/O benchmark.

- First, using NDP-MCP (Figure 1), we stage-in datasets from the National Data Platform with the prompt: *"Find the latest dataset of the earthscope organization and download it"*.
- Second, using the ADIOS-MCP agents explored the output data of a LAMMPS application simulating the melting of a face-centered cubic gold crystal. Given the prompt: *"Explore the dataset and plot the trajectory of a single atom over time, make the atom ID a parameter in the script"*, the agent generated Figure 2.
- Third, using Jarvis-MCP, agents were capable of deploying the IOR benchmark through the prompt: *"Deploy ior with 256 processes on 16 nodes using the scratch filesystem"*. Compared to the 5-10 minutes required for manual setup Figure 3 shows the execution time and success rate of different models and agents.

References

[1] Anthropic. 2024. Model Context Protocol specification. GitHub. https://github.com/anthropics/mcp Accessed: August 1, 2025.

[2] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374* (2021).

[3] Shusen Liu, Haichao Miao, and Peer-Timo Bremer. 2025. ParaView-MCP: An Autonomous Visualization Agent with Direct Tool Use. arXiv:2505.07064 [cs.HC] https://arxiv.org/abs/2505.07064

[4] OpenAI. 2023. ChatGPT plugins. https://openai.com/index/chatgpt-plugins/. Accessed: August 1, 2025.

[5] Path Integral Institute. 2025. MCP.Science. https://github.com/pathintegral-institute/mcp.science. Version 0.1.0, released 2025-05-21.