Agentic Search Efficiency for Scientific Data Discovery

Shazzadul Islam Jaime Cernuda Xian-He Sun Anthony Kougkas
sislam6@hawk.illinoistech.edu {jcernudagarcia, sun, akougkas}@illinoistech.edu

Introduction

Scientific data search demands reasoning that does not scale: inspecting what each dataset contains, converting between unit conventions, refining queries when results fall short. AI agents can automate this, but a single query can burn nearly a million tokens before producing an answer.

We present **CLIO Search**, a retrieval engine that implements an *observe–decide–act–evaluate* cycle for scientific data discovery:

1. **Discover (Observe):** profile each corpus (document count, measurements, metadata density, unit domains) to identify viable search strategies.
2. **Reason (Decide):** select retrieval branches based on what the corpus provides — skip scientific if no measurements, skip vectors if no embeddings.
3. **Transform & Search (Act):** execute four parallel branches — BM25, dense vector (MiniLM-L6-v2), graph traversal, and scientific operators (SI conversion across 58 units in 12 domains + formula normalization) — then merge with a scientific hard filter.
4. **Execute (Evaluate):** expand, narrow, pivot, or stop; rewritten queries re-enter the pipeline.

The agent receives compact ranked citations, not raw catalog JSON. Across 10 discovery queries on the National Data Platform (341 NOAA / NASA / ECMWF datasets) using `claude-sonnet-4-20250514`, a CLIO-backed agent reduces tokens by **52%** and wall time by **91%** versus a native Claude agent — answering all 10 queries correctly.

System Architecture

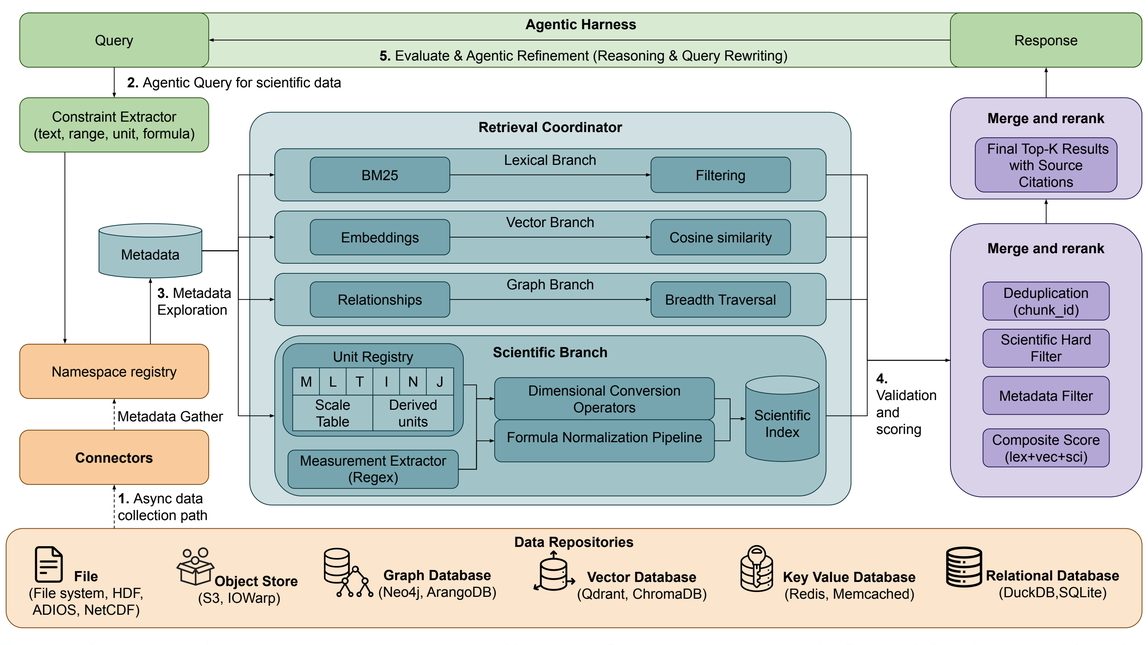

Figure 1: Connectors index heterogeneous repositories; the Namespace Registry routes queries; the Retrieval Coordinator runs lexical, vector, graph, and scientific branches in parallel; the agentic loop refines queries iteratively.

IOWarp Storage Scaling

CLIO profiles IOWarp CTE blob stores before querying; profiling cost scales *sub-linearly* with blob count, so even very large object stores remain cheap to inspect.

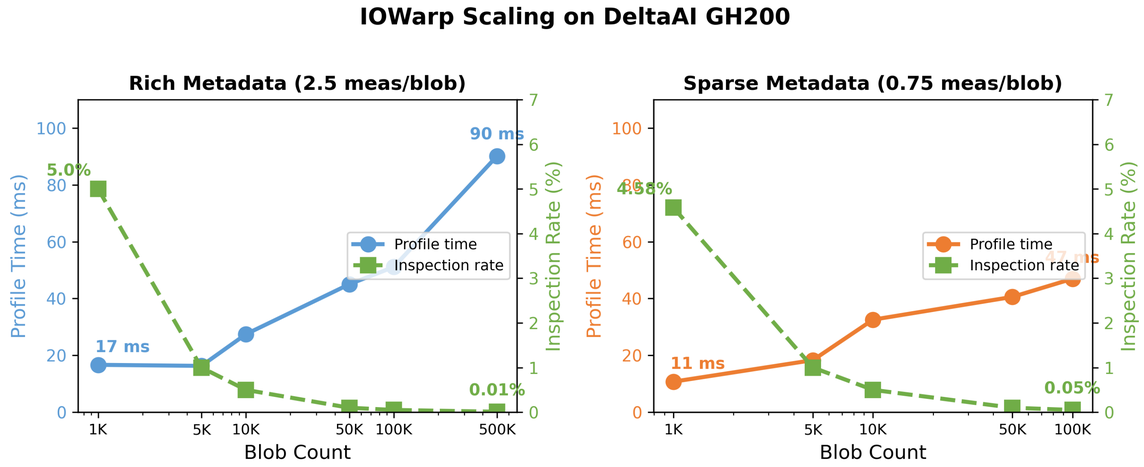

Figure 2: Profiling on DeltaAI GH200, 1K → 500K blobs. **Rich metadata** (2.5 meas/blob): **17 ms** at 1K → **90 ms** at 500K; inspection rate 5.0% → 0.01% (500× less content scanned). **Sparse metadata** (0.75 meas/blob) follows the same sub-linear curve, validating CLIO's adaptive strategy on heterogeneous HPC blob storage.

Takeaway: CLIO indexes scientific metadata at ingest, so retrieval is content-aware — not just tag- or filename-based. **Profiling scales sub-linearly:** 500× data growth (1K → 500K blobs) costs only 5.3× more time (17 → 90 ms), so per-query corpus inspection stays cheap at HPC scale.

Three-Path Experiment

We measure the *cost* of agentic scientific search by holding everything constant except the retrieval backend. The same LLM (`claude-sonnet-4-20250514`) answers the same 10 discovery queries (temperature, pressure, wind, radiation, soil moisture) over the same 341 National Data Platform datasets — three times, once per backend:

- **Claude Native — no domain retrieval.** Web search and sub-agents only; the agent must locate datasets through general web tools.
- **NDP-MCP — raw catalog access.** Catalog search via Model Context Protocol returns *raw dataset JSON*; the agent parses, filters, and ranks in-context.
- **CLIO — science-aware retrieval.** Datasets are indexed and profiled ahead of time; queries return *ranked citations* (∼2K tokens) the agent can use directly.

Results: Same Accuracy, Dramatically Different Cost

All three paths answer **10/10 queries correctly**:

Method	Tokens	Token △	Time △	Correct
Claude Native	989,417	—	—	10/10
NDP-MCP	988,954	−0.05%	−65.7%	10/10
CLIO	**474,101**	**−52.1%**	**−91.3%**	**10/10**

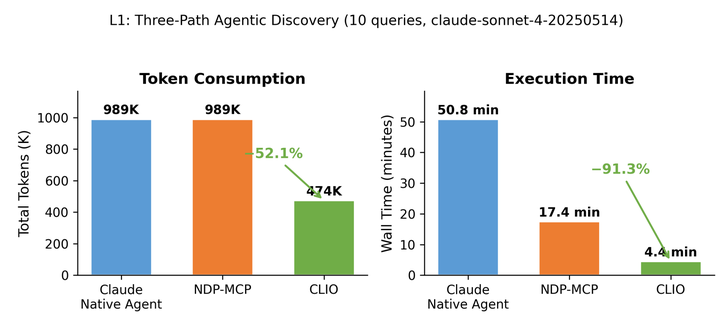

Figure 3: Total token consumption (left) and wall time (right) across 10 queries: 52% token and 91% time reduction over the Claude Native baseline.

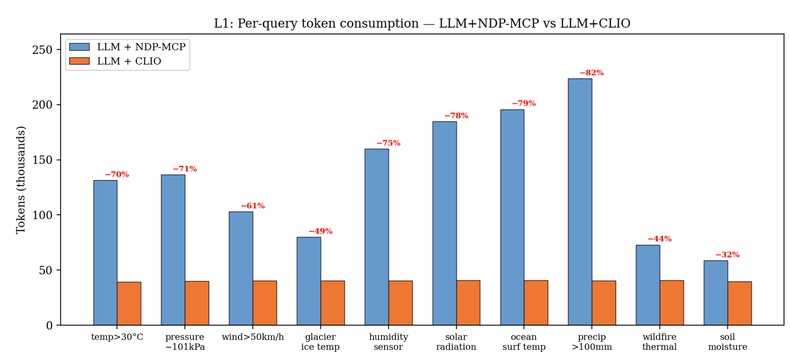

Figure 4: Per-query token consumption. NDP-MCP (blue) consumes 60K–225K tokens because the agent parses raw JSON in-context; CLIO (orange) is flat at ∼40K. Reductions span −32% to −82% — the saving holds across query types, not only on the average.

Broader CLIO Results

Beyond agentic efficiency, across controlled benchmarks, real observatory data, and distributed HPC:

- **Cross-unit retrieval:** 0.99 P@5 vs. 0.40 for BM25 on 210 docs / 80 queries / 7 domains
- **Real-world NOAA:** +87% precision over hybrid on 288 GHCN-Daily records
- **Distributed scale:** p50 < 42 ms on 2.5M arXiv across 4 GH200 nodes (NCSA DeltaAI), 0.94 weak-scaling efficiency
- **Federation:** 1.0 routing recall across 100 namespaces while skipping 52% of irrelevant ones

Conclusion

Scientific data discovery requires search that *reasons*, not just retrieves. Three gaps make automation hard: **dimensional blindness** (no arithmetic for unit conversion), **metadata heterogeneity** (no adaptive strategy per corpus), and **storage fragmentation** (no federation across HPC backends). CLIO closes all three by separating *reasoning* from *execution* in an observe–decide–act–evaluate cycle: SI conversion, profiling, and ranking happen *before* the agent sees results. The lesson is that agentic search is a **co-design** problem — the bottleneck is not LLM capability, but the shape of the data the agent is forced to interpret.

Acknowledgments

This work is part of the CLIO Kit project within the IOWarp platform.

CLIO

Illinois Institute of Technology
sislam6@hawk.illinoistech.edu

IOWarp Community