Data Lake vs Lakehouse vs Data Warehouse: A Comprehensive Comparison in HPC

Erdni Mankirov

GWDG, University of Goettingen

July 2, 2025

Outline

What This Topic Means

- Investigating how different storage/query models handle real-world HPC datasets.
- Focusing on flexibility, performance, reliability, and operational overhead.
- Assessing end-to-end workflows: ingestion, analytics, fault recovery.
- Delivering actionable guidance for scientific computing centers managing petabyte-scale data.

Heterogeneous HPC Data Types

- **Structured:** CSV tables with simulation metrics (time series, scalar outputs).
- **Semi-Structured:** JSON logs containing runtime parameters, error traces.
- **Unstructured:** HDF5/NetCDF files storing multi-dimensional arrays (e.g., climate grids, spectral data).

Rationale for heterogeneity:

- Real-world HPC pipelines produce mixed formats in a single workflow.
- Data Lake/House designed for multi-format, schema-on-read ingestion.
- OLAP engines (ClickHouse/Greenplum) optimize structured queries but require ETL for other types.

What is a Data Lake?

- Centralized repository for all types of raw data: structured, semi-structured, unstructured.
- Schema-on-read: data interpretation happens at query time.
- Built on object storage (e.g., MinIO, S3) for high scalability and low cost.

What is a Data Warehouse?

- Structured storage optimized for analytics (OLAP) with rigid schemas (schema-on-write).
- ACID transactions, indexing, and query optimization for consistent performance.
- Ideal for business intelligence, reporting, and complex SQL workloads.

What is a Data Lakehouse?

- Hybrid architecture combining Data Lake flexibility and Data Warehouse reliability.
- Provides ACID transactions on object storage via Delta Lake or Iceberg.
- Supports both BI/SQL and data science/ML workflows (Spark engine).
- Features: time travel, unified metadata, data compaction, upserts.

What?

- Deploy and configure three storage/query architectures on GWDG via SSH and Slurm.
- Generate heterogeneous HPC datasets (CSV, JSON, HDF5) for each environment.
- Execute benchmarks to measure throughput, latency, metadata overhead, recovery, and storage efficiency.

Why?

- Address a research gap: no direct comparisons of Lake, Lakehouse, and OLAP engines for HPC data.
- Provide GWDG with data-driven recommendations for optimal storage/query paradigms.
- Highlight trade-offs in flexibility, performance, reliability, and resource utilization.

How?

1. **Week 1:** Literature review and script preparation (remote).
2. **Week 2:** SSH deployment of Data Lake, Lakehouse, and Data Warehouse stacks.
3. **Week 3:** Data generation scripts and ingestion pipelines (ELT/ETL).
4. **Week 4:** Execute bulk write/read and OLAP benchmarks (fio, Spark, native clients).
5. **Week 5:** Conduct fault injection, scale-out testing, and metric collection.
6. **Week 6:** Data analysis, visualization, and final report/slides preparation.

System Architectures

- **Data Lake:** MinIO + Apache Iceberg + Trino
- **Data Lakehouse:** MinIO + Delta Lake/Iceberg + Apache Spark
- **Data Warehouse:** ClickHouse or Greenplum

GWDG Resources Used:

- 6–8 compute nodes via SSH
- 32 cores, 128 GB RAM, 1 TB SSD per node
- Slurm job scheduling, Prometheus/Grafana monitoring
- Singularity containers for isolated software deployment

Deployed on GWDG via SSH + Slurm for controlled benchmarking.

Key Performance Metrics

Throughput Aggregate MB/s for concurrent reads/writes using fio and native tools.

Latency Percentile latencies (P50, P95) for single-record operations in SQL engines.

Metadata Overhead Time to register schemas, commit transactions, and list partitions.

Recovery Time Time to restore service after node failure.

Storage Efficiency Ratio of user data volume to total consumed storage (including metadata).

Why These Metrics?

- Throughput Latency: assess raw performance and responsiveness
- Metadata Overhead: measures operational cost of schema management
- Recovery Time: evaluates resilience under failures
- Storage Efficiency: quantifies overhead vs usable data capacity

Required Resources

- 2–3 nodes per architecture: 32 cores, 128 GB RAM, 1 TB SSD.
- Slurm scheduler access for fio, Spark, Trino, ClickHouse, Greenplum jobs.
- Prometheus + Grafana for metrics collection.

Workflow Overview

1. **Provisioning Environments**
   Deploy MinIO, Spark, Trino, ClickHouse, Greenplum via scripted automation.
2. **Data Generation**
   Synthesize 10, 20, 30 GB datasets (CSV, JSON, HDF5).
3. **Data Ingestion**
   ELT for Lake/Lakehouse, ETL for DWH using COPY or native bulk loaders.
4. **Benchmark Execution**
   - Bulk write/read: fio, Spark write jobs, Trino/ClickHouse reads.
   - OLAP queries: aggregations, joins, window functions.
   - Mixed workloads: concurrent ingestion and analytics.
5. **Fault Injection**
   Simulate node crashes, measure failover and recovery.
6. **Analysis**
   Python/Matplotlib for plotting throughput vs size and latency CDFs.

Scope Optimization

- Focus on micro-benchmarks (10–30 GB) to reduce resource usage.
- Derive scaling trends rather than testing petabyte volumes.
- Prioritize read-heavy OLAP scenarios common in HPC analytics.
- Extrapolate larger-scale performance using linear models and Amdahl's Law.

Feature Comparison

Feature	Data Lake	Lakehouse	Data Warehouse
Storage	Object store	Obj store + tables	Block/Relational
Schema Model	On-read	Hybrid	On-write
ACID Support	No	Yes	Yes
Query Engine	Trino	Spark SQL	Native SQL
OLAP Performance			
Flexibility			
Metadata Control	Moderate	Strong	Strong
Recovery Speed	Fast	Fast	Moderate

Outcomes & Impact

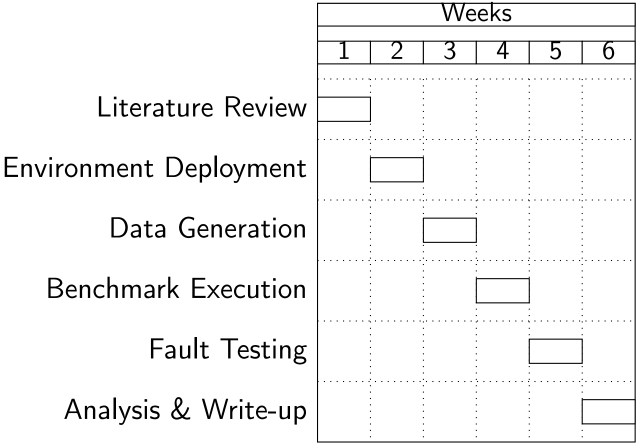
- Detailed performance profiles for three storage paradigms.
- Informed recommendations for GWDG storage configurations.
- Resource-efficient methodology applicable to future evaluations.
- Enhanced decision-making for HPC data management strategies.

Key Literature References

- "An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management"
  - Highlights metadata governance challenges and schema trade-offs.
  - Advocates hybrid models to balance flexibility and consistency.
- "Data Lakes: A Survey of Functions and Systems"
  - Surveys core Data Lake functions: ingestion, storage layers, processing.
  - Emphasizes critical role of metadata catalogs to prevent "data swamps."
- "An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management" (ResearchGate, 2022)
  - Reviews enterprise integration patterns and real-world deployments.
  - Identifies performance and governance considerations at scale.

Links:
https://www.mdpi.com/2504-2289/6/4/132
https://ieeexplore.ieee.org/document/10107808
https://www.researchgate.net/publication/365240520$_{A}n_{O}verview_{o}f_{D}ata_{W}arehouse_{a}n$

Gantt Chart


Weeks
1
2
3
4
5
6
Literature Review
Environment Deployment
Data Generation
Benchmark Execution
Fault Testing
Analysis & Write-up

Questions

Thank you for your attention!
Any questions?