MetrANOVA

An Open Source Software Stack for Sharing Network Telemetry

Andy Lake - ESnet

5th European perfSONAR User Workshop - May 6th, 2026

ESnet GÉANT INDIANA UNIVERSITY INTERNET2 TACC

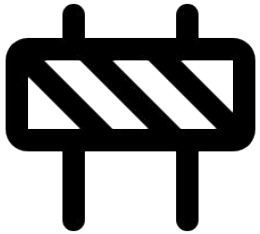
What is MetrANOVA?

Open source software stack for collecting and storing **flow**, **SNMP** and **streaming telemetry** data

Key Features:

- **Free** and **open source** always
- **Easy to deploy** at small and large scales
- Easy to integrate **metadata from multiple sources**
- Hooks to **share data responsibly** between organizations

MetrANOVA's Mission



1. **Lower the barriers of entry** to get relevant network measurement data

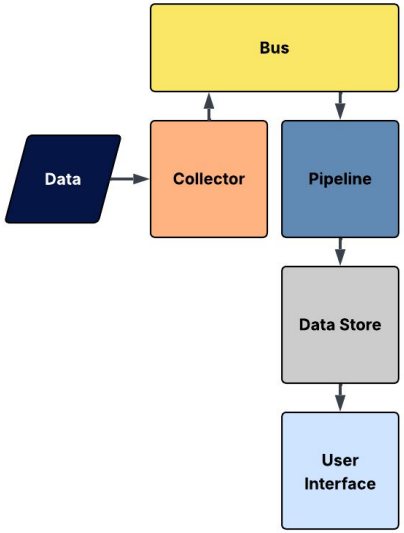
2. **Advocate for end-to-end network measurements** with appropriate access within all of R&E

Consortium Participants *(alpha order)*

Active Workstreams

1. Develop Network Measurement Software Stack
   a. Lowerer barrier to entry for both big and small deployments
   b. Leverage prior work with NetSage, Stardust, etc
   c. Data Store and Ingest Pipeline primary focus at this stage
      i. Elasticsearch and Logstash used in past
      ii. Both very effective but resource intensive
      iii. Need something more thrifty and performant

2. Develop a data sharing policy framework
   a. Support goal of enabling end-to-end visibility
   b. Provide requirements for stack developers



Network Measurement Stack

1. Collect data
2. Normalize and enrich
3. Store for retrieval
4. Analyze and present

2025 Data Store Evaluation

The Goal: Selecting a Datastore

Requirements *(big ones)*:

- Must support port and host stats
  - Low cardinality metadata
  - Consistent timestamps
- Must support flow data
  - High cardinality 5 tuple
  - Sporadic/unpredictable timestamps
- Must not be proprietary
- Must scale out to multiple nodes
- Should provide SQL like query interface
- Should provide fine-grained access control

Approach:

- Conduct Literature review and select handful of contenders
- Test on different size nodes
- Evaluate insert and query performance
- Tests conducted in Google Cloud Platform
  - Used VMs
  - SSD backed persistent disks
- Three scenarios
  - Flow Data
  - Port data with small set of fields
  - Port data with large set of fields
- Focus on progress over perfection

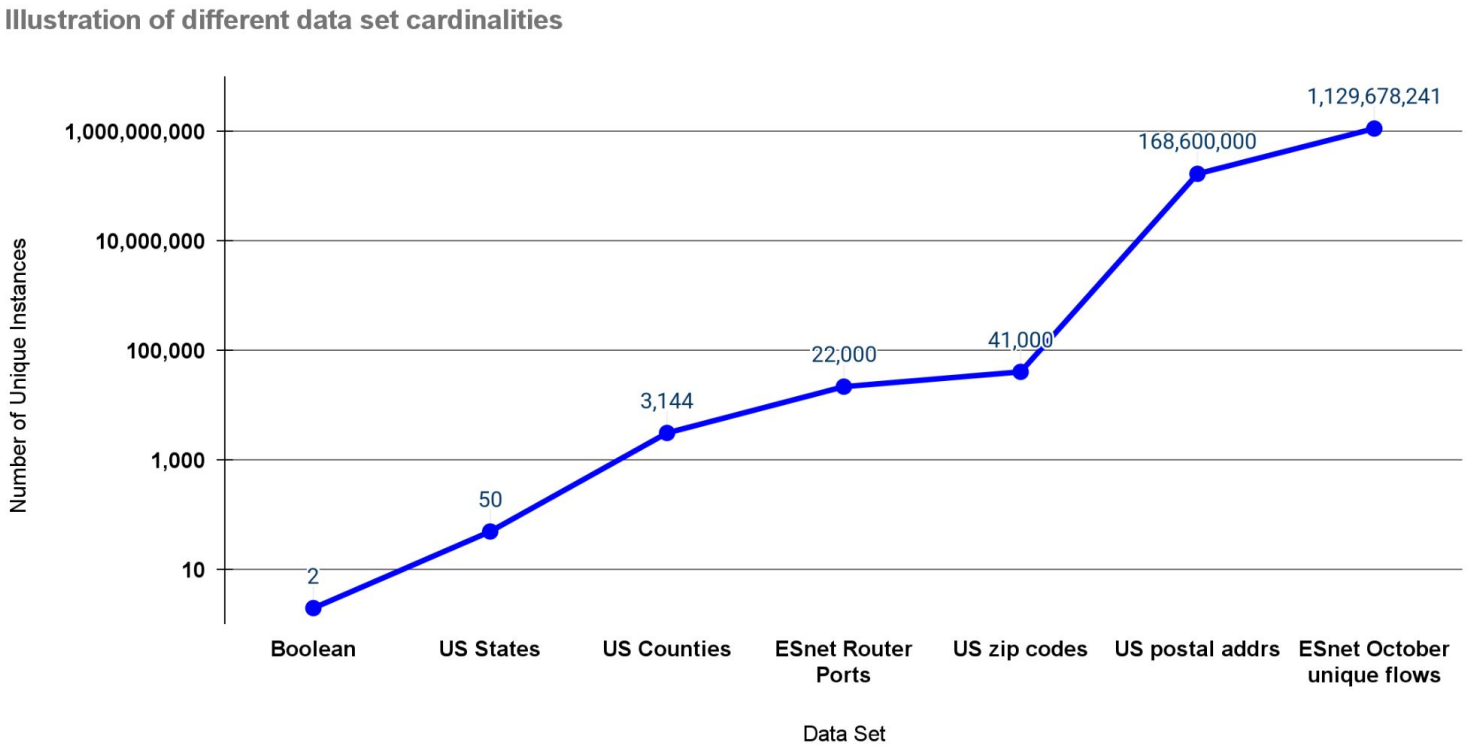
Data Store Evaluation: The Contenders

Why These Contenders?

- Based on literature review of availability systems that meet basic requirements
- Based on what we knew was in use in the community.
- We might have skipped over some of your favs
- Largely, these choices were driven by flow data cardinality

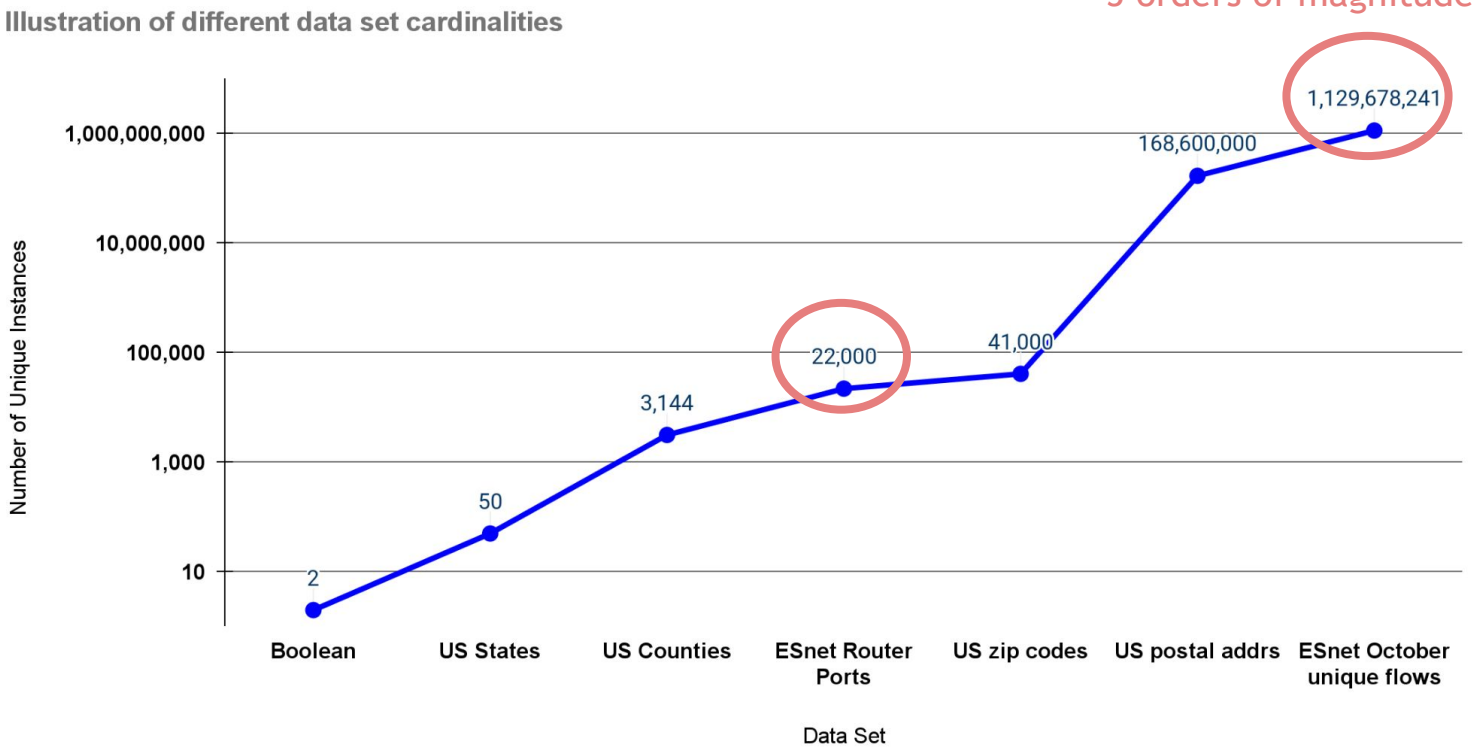
Flow Data Cardinality: a gating factor

Limiting factor: High Cardinality



Limiting factor: High Cardinality

5 orders of magnitude increase



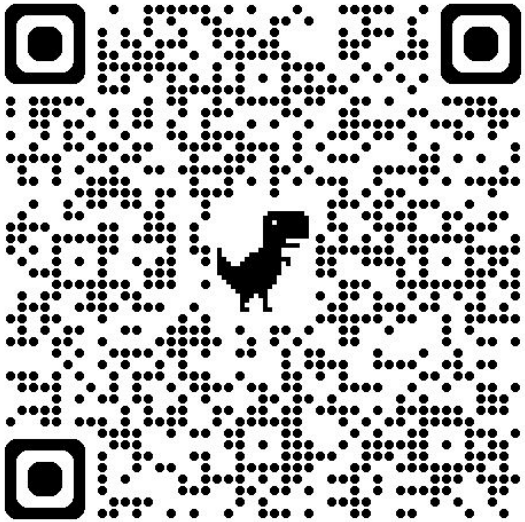
Limiting factor: Flow Data

- Two fundamental challenges
  - high cardinality
  - Inconsistent / sparse flows with same N-Tuple

- Common options that don't survive the combo
  - Prometheus
  - InfluxDB
  - Victoria Metrics

Detailed Evaluation Results

- Performed data store evaluation using representative data sets of snmp interface and flow data
- Compared databases suitable for high-cardinality time series data:
  - ClickHouse
  - ElasticSearch
  - MongoDB
  - OpenSearch
  - TimescaleDB
- Initial findings public in linked document
- Final Choice: **ClickHouse**

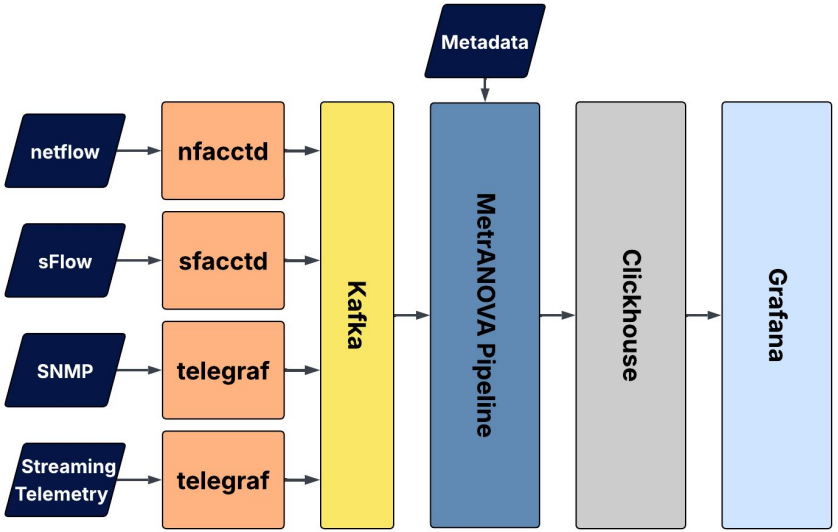


Link to results document

MetrANOVA Software Stack

Building the MetrANOVA Stack

- Support common network measures such **netflow, sFlow, SNMP** and **Streaming Telemetry**
- Easy to deploy package
- Support for custom metadata
- Pre-built dashboards for common use cases
- Containerized deployment



MetrANOVA @ SC25

Booth Map

Last 6 hours | Refresh 30s

Which booths are using the most data?

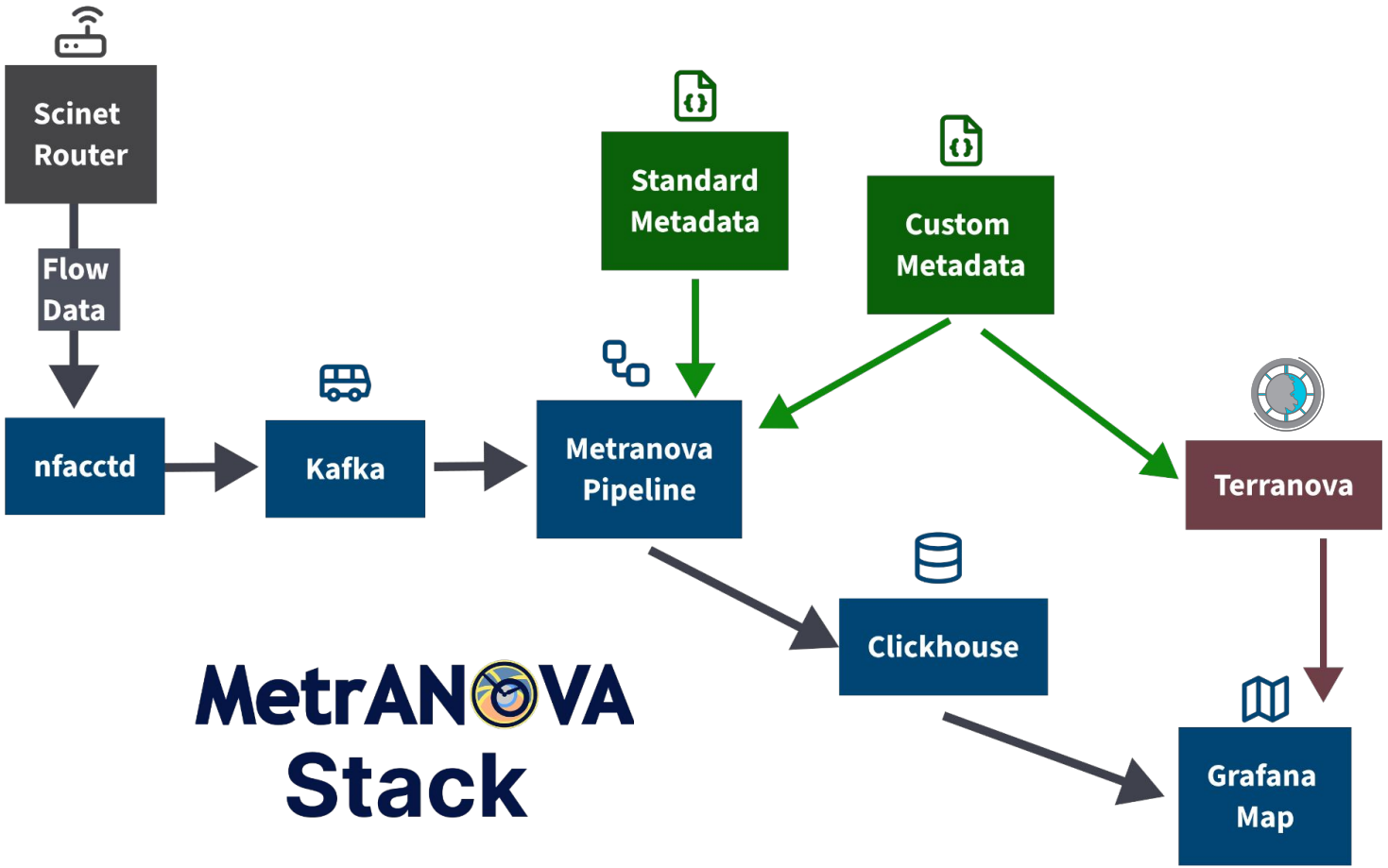
Top Booths by Volume (Incoming)

Booth	Organization	Volume
Booth 3224	California Institute of Technology /CACR	359 GB
Booth ES34	Dell Technologies	30.1 GB
Booth 3131	StarLight	20.5 GB
Booth 2141	MathWorks	20.0 GB
Booth 1627	Microsoft	19.1 GB
Booth 2207	AWS	19.0 GB
Booth 3840	Purdue University	12.5 GB

Top Booths by Volume (Outgoing)

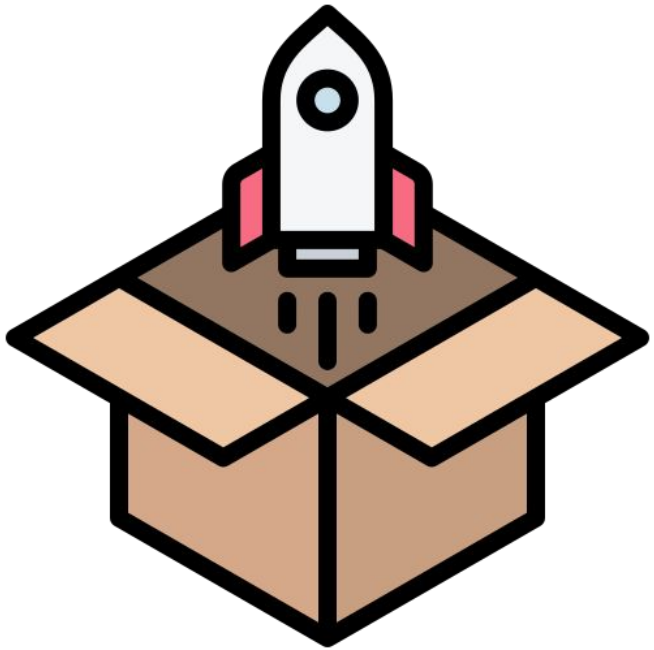
Booth	Organization	Volume
Booth 3224	California Institute of Technology /CACR	636 GB
Booth ES47	Weka	18.7 GB
Booth 2207	AWS	13.0 GB
Booth 3131	StarLight	6.04 GB
Booth 726	HLRS	3.05 GB
Booth 3814	Lenovo	1.64 GB
Booth ES46	Lightmatter	1.37 GB

Powered by MetrANOVA


Scinet Router
Flow Data
nfacctd
Kafka
Standard Metadata
Custom Metadata
Metranova Pipeline
Terranova
Clickhouse
Grafana Map
MetrANOVA Stack

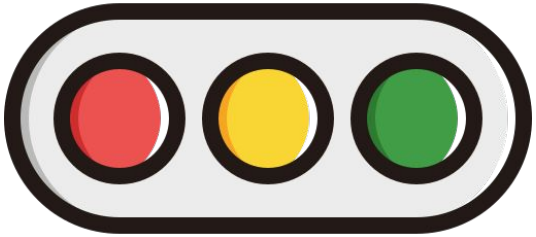
MetrANOVA distribution

- Expect to release initial beta version in **early 2026**
- It will be available as a set of **docker containers** and **k8s helm charts**
- Guides to help with larger deployments
- Communication channels to support users



Extending End-to-End

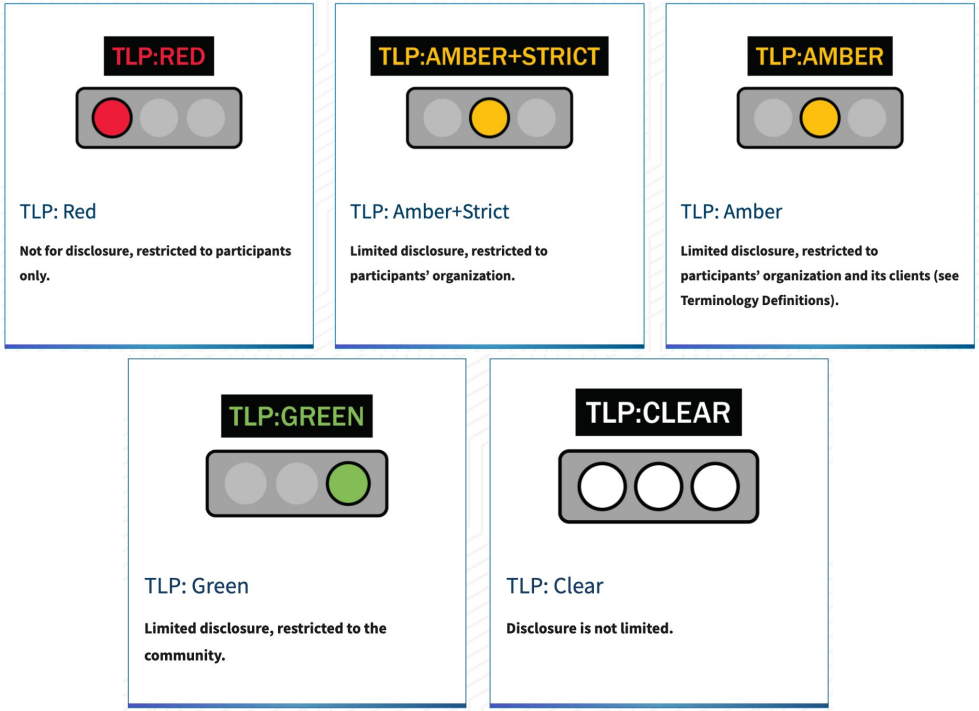
Different Kinds of Sharing



- **Different data products have different levels of sensitivity** - A throughput measurement is not the same as raw flow data with customer IP addresses

- **Clear policies help set expectations** - We need well understood terminology to describe the policies we use to share data.

Signaling Sensitivity



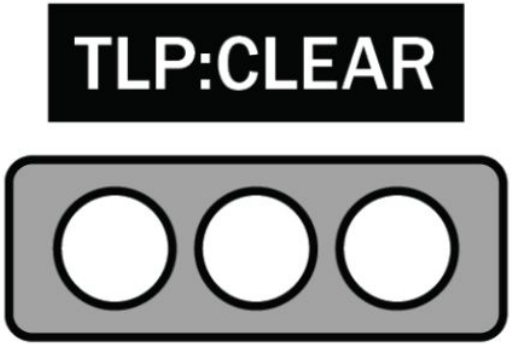
Traffic Light Protocol provides an pre-defined and relatively easy way to talk about policy

Each measurement has a **level** and a **scope**

Defines primitives but still **open questions for enforcing** those primitives

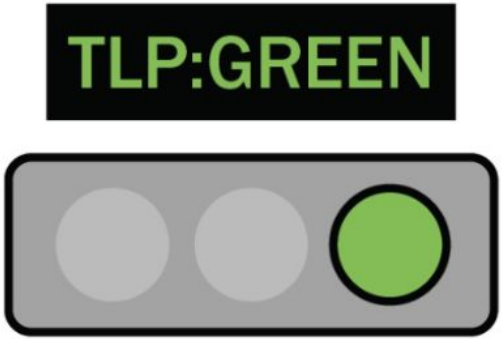
https://www.cisa.gov/news-events/news/traffic-light-protocol-tlp-definitions-and-usage

Example Policy Buckets



(no login, public data)

perfSONAR measurements

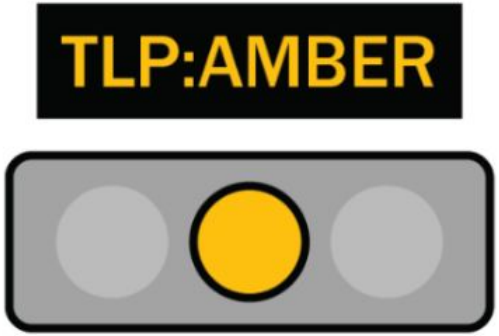


(any R&E login)

Aggregate flow data, interface data

ESnet GÉANT INDIANA UNIVERSITY INTERNET2 TACC

Example Policy Buckets

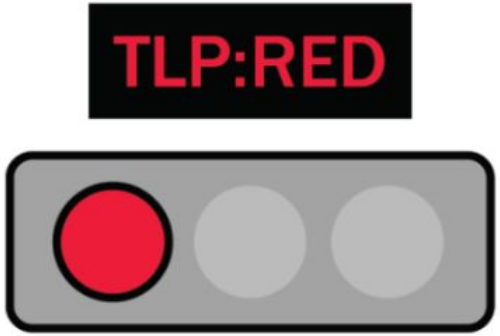


(login with specific permissions)

Edge flow data with full IPs.

Examples:

SCOPE AS, SCOPE L3VPN

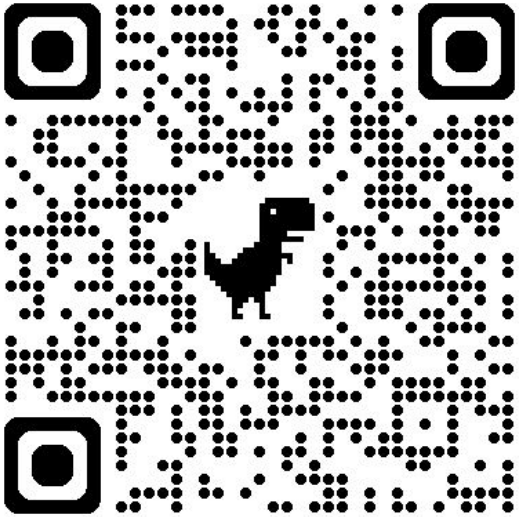


(not exported to shared instance)

Core flow data, highly sensitive subsets of other data

Data Sharing Considerations Document

- Drafting document that covers considerations when it comes to sharing network data
- Topics covered such as labeling, classification and enforcement
- Goal is to not be overly prescriptive while still providing useful guidance



Link to policy document

Roadmap: What Comes Next

Build production deployments at IU GlobalNoc and ESnet 2026

Ship software beta release in 2026

Documentation, guides and workshops coincide with software release

Website

Mailing List

Questions?

ESnet GÉANT INDIANA UNIVERSITY INTERNET2 TACC

Testing the Data Stores

Evaluation Methodology: Standard Node Sizing and I/O Layer

Node Sizing:

- **Small**: 2 CPU / 16GB / Dedicated SSD
- **Medium**: 4 CPU / 64GB / Dedicated SSD
- **Large**: 16 CPU / 256GB / Dedicated SSD

Evaluation Methodology: Ingest Testing

Data Sets:

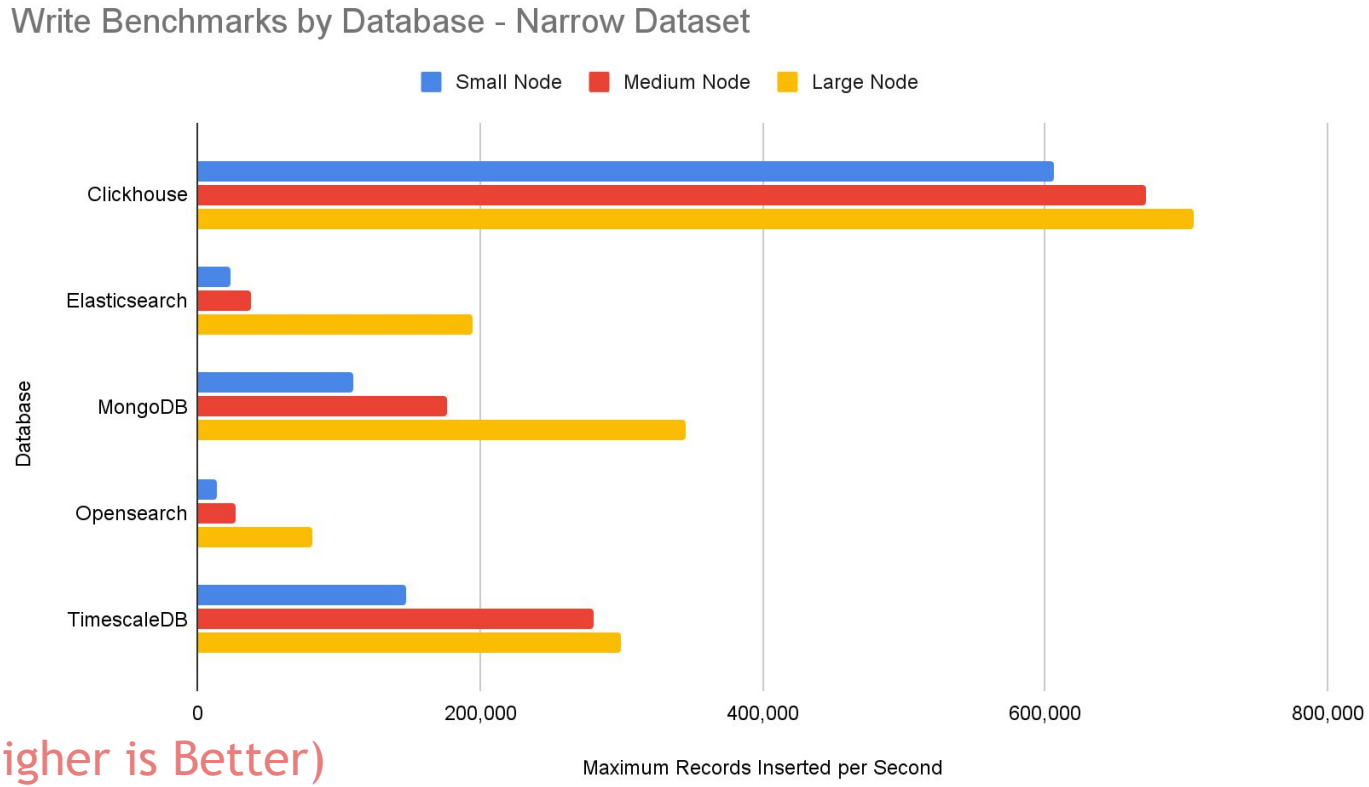
- **Narrow**: Low Field count, Medium Cardinality
- **Wide**: High Field count, Medium Cardinality
- **Flow**: Low Field Count, High Cardinality

Observation:
Data order is important

For some data stores, during ingest, data is batched on disk (and in RAM) in chunked groups.

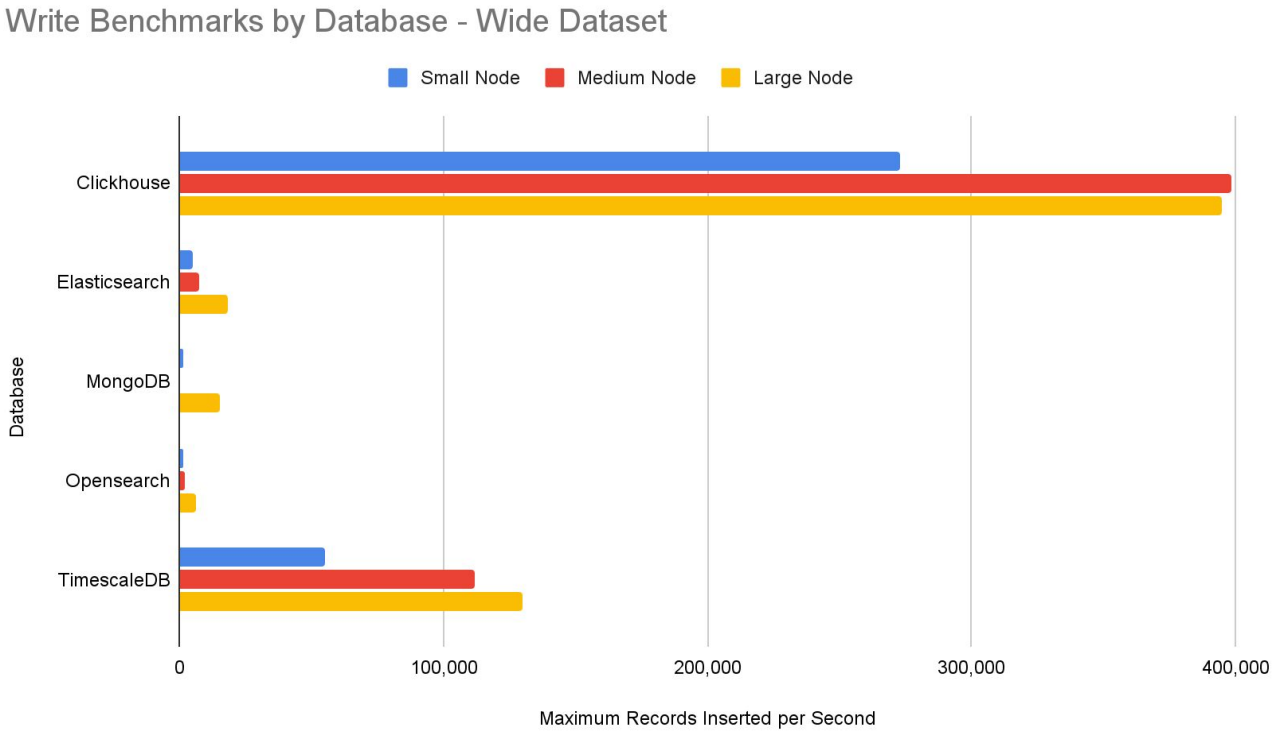
While a specific group is under write by an ingest worker, another worker cannot acquire a lock. We were using many simultaneous writer processes in testing. Ordering data could significantly influence ingest speeds (by orders of magnitude).

Results: Ingest Testing



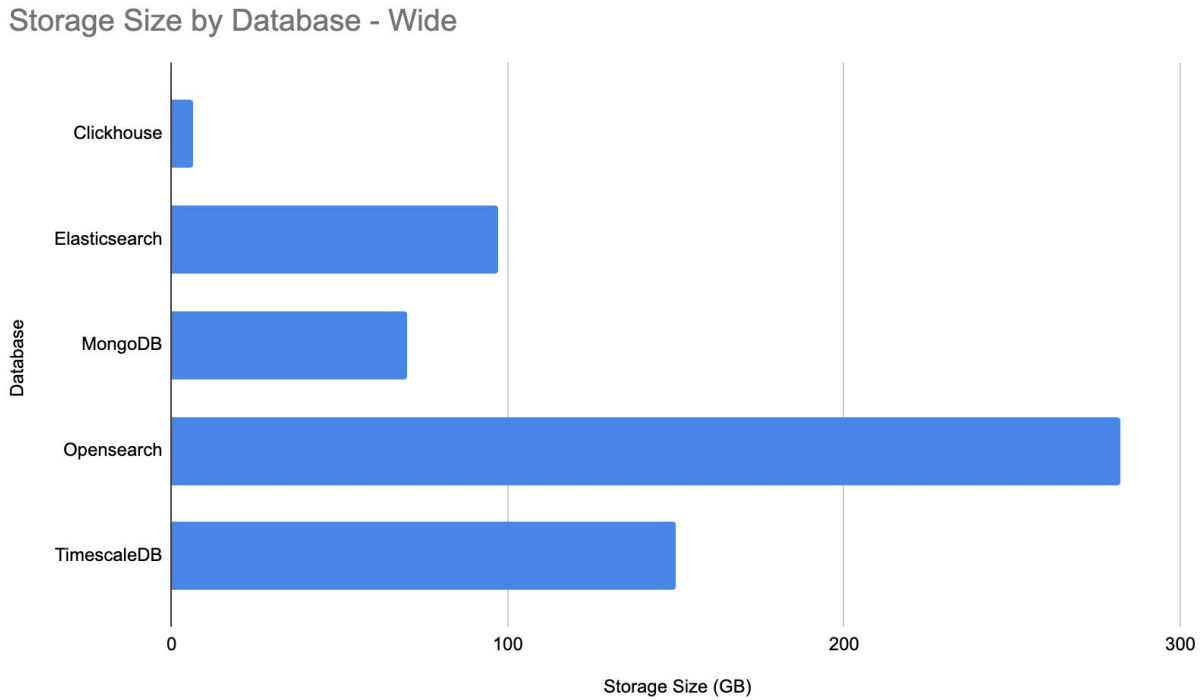
(Higher is Better)

Results: Ingest Testing



(Higher is Better)

Results: Stored size on Disk



(Smaller is Better)

Methodology Changes After First Round

Node Sizing

Only Use ‘Medium’ Nodes

(They deliver a good approximation of performance)

Data Sets

Ignore ‘Narrow’ Data

(‘Narrow’ Dataset was below the performance ‘knee’)

Narrow the Field

Drop MongoDB, OpenSearch from consideration

OpenSearch: Slower than ES

MongoDB: did not complete testing

Evaluation Methodology: Read Benchmarking

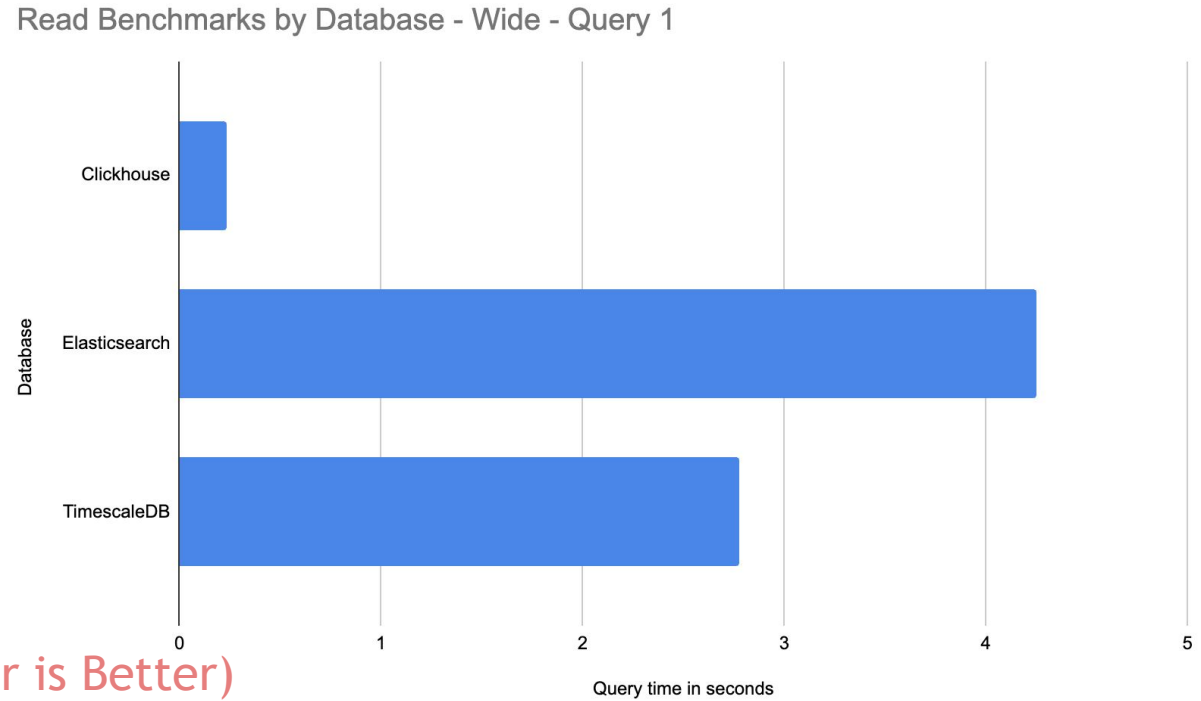
Data Sets:

- Wide: High Field count, Medium Cardinality
- Flow: Low Field Count, High Cardinality

Query based read use cases:

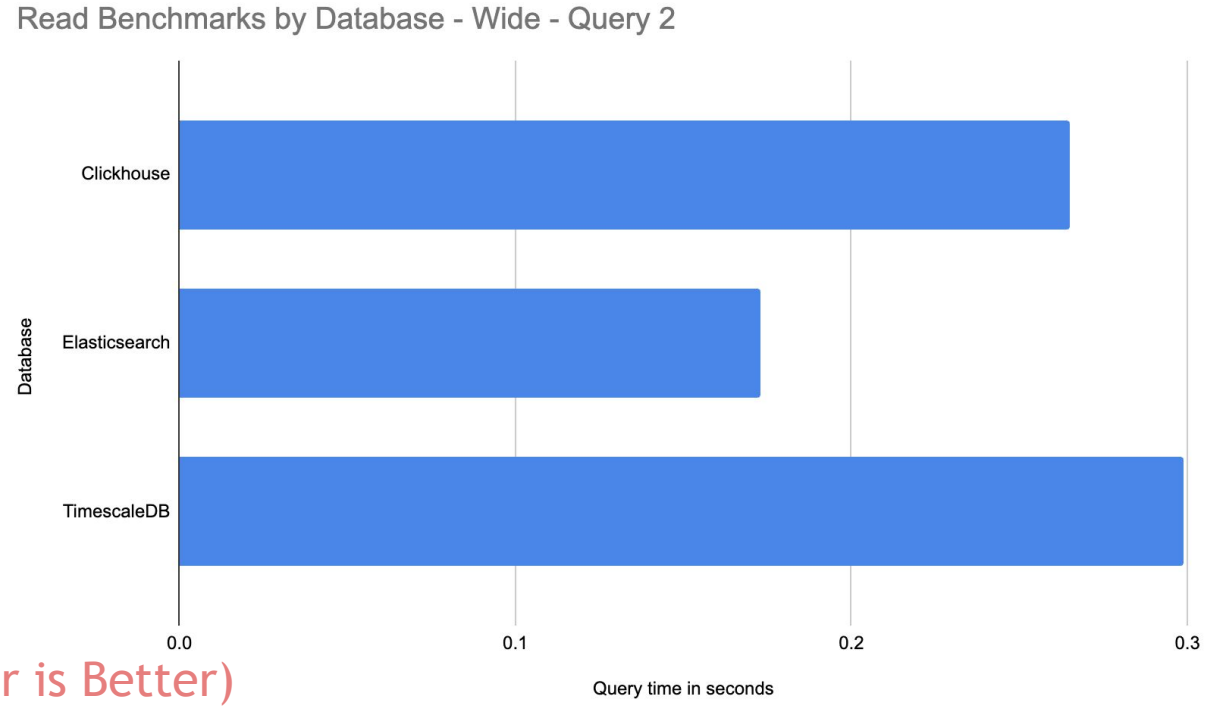
- Based on cases in production today
- Needle in haystack, metadata based summaries, etc

Evaluation Methodology: Read Benchmarking Results



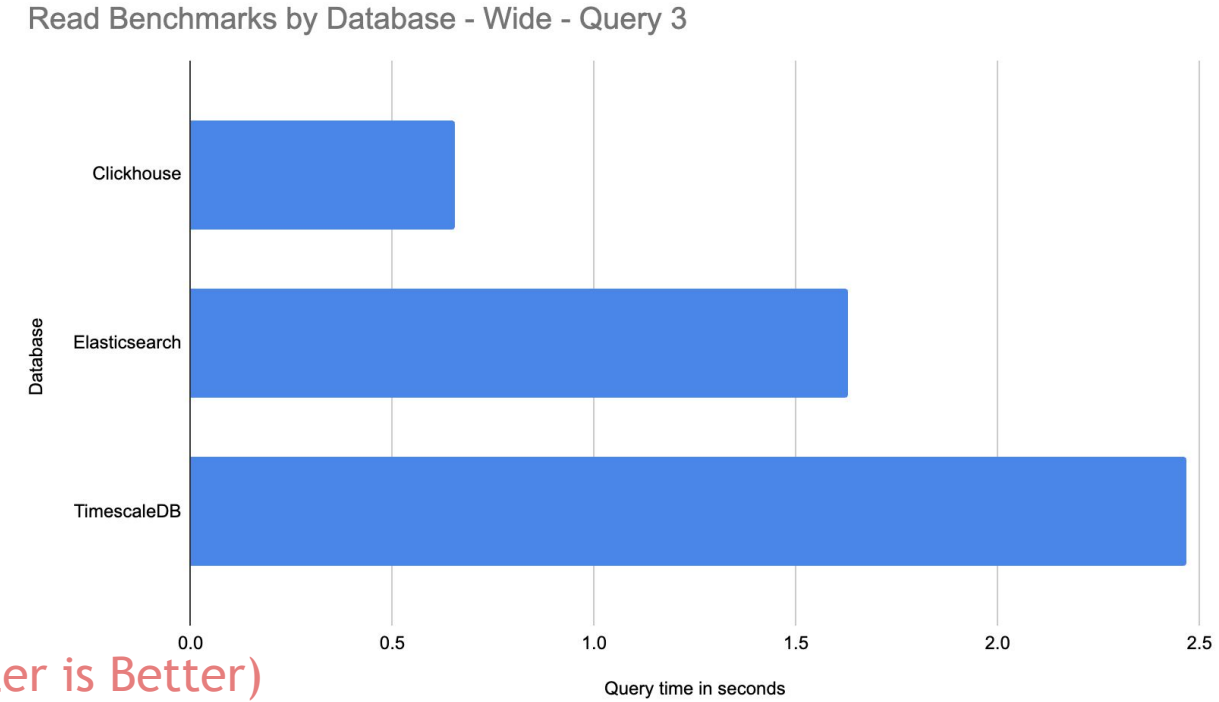
(Smaller is Better)

Evaluation Methodology: Read Benchmarking Results



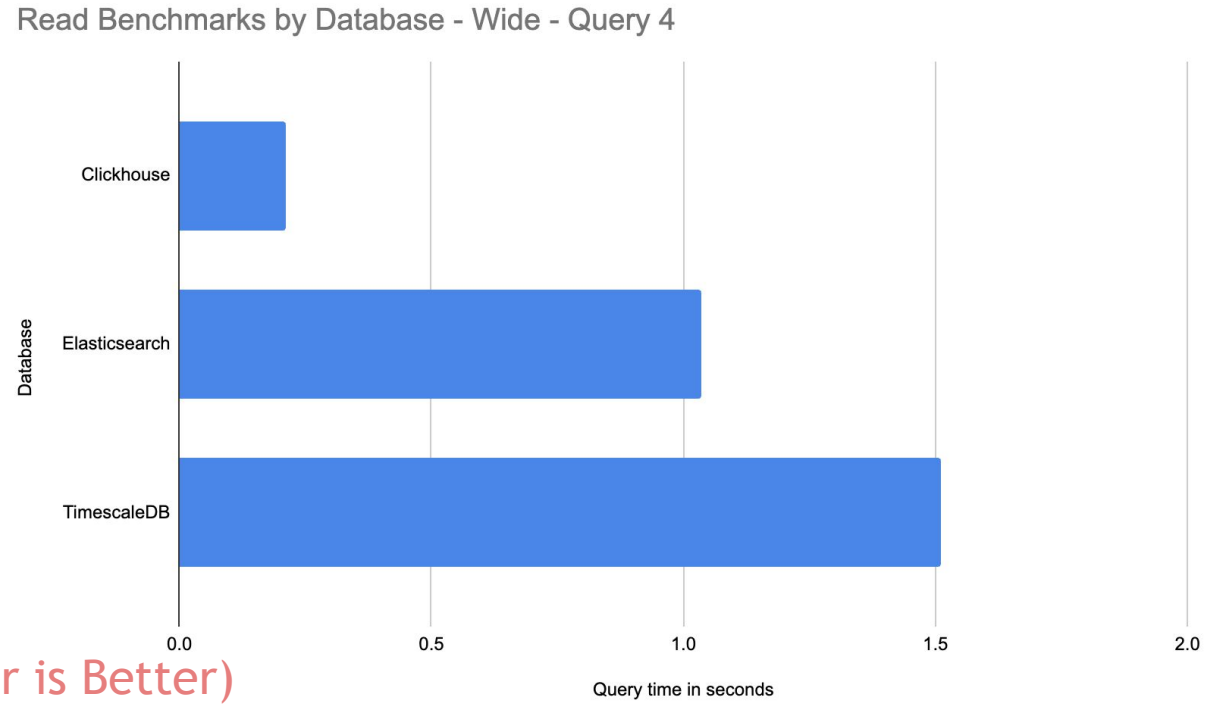
(Smaller is Better)

Evaluation Methodology: Read Benchmarking Results



(Smaller is Better)

Evaluation Methodology: Read Benchmarking Results



(Smaller is Better)

Testing Results Summary

Write Speed

Winner: Clickhouse

Read Speed

Winner: Mixed

(Elastic and Clickhouse)

Data Storage Efficiency

Winner: Clickhouse

Outcomes:

1. We have selected Clickhouse as our Datastore
2. A full software stack based on this choice will be available early 2026