

HarperDB

VS. Macrometa*

38% of devs say they are more likely to use a database that is on-prem and cloud.

- 451 Research

Overview

Based on functional requirements, the ideal use cases for HarperDB and Macrometa may look nearly identical. But those are only the functional requirements. Once you consider the real-world impact of developer experience, throughput, latency, scalability, and TCO (non-functional requirements), the use case overlap evaporates.

HarperDB handles 140,000 transactions per second (TPS) per node and scales linearly.

Macrometa can handle 300 TPS globally across their entire 170 node network. In comparison, HarperDB on 170 nodes can handle 23,800,000 TPS or **79,300x the transaction volume of Macrometa** 🤖.

Such high performance on a single node means companies need significantly fewer servers, **saving enterprises millions per year.**

HarperDB is a future-proof investment. With HarperDB, you have the flexibility of a fully managed serverless FaaS architecture with the flexibility to deploy anywhere. Macrometa restricts data storage and compute to the cloud, limiting your technology runway. In a world moving towards a Web3 paradigm, today's technology investments must be able to adapt. HarperDB's platform allows enterprises to shift seamlessly between the cloud, edge, and decentralized peer-to-peer networks, giving you a runway to the future.

Keep developers happy and develop solutions in less time.

Macrometa is particular. They require developers to learn their proprietary hybrid SQL -Java language to write queries. They need developers to rework their CI/CD pipeline, adjust their version tools, and work around limitations on NPM packages.

On the other hand, HarperDB simplifies existing workflows. With HarperDB, you get to keep using your existing CI/CD pipeline, GitHub actions, NPM packages, IDE, and more. Oh, and want to write a query? Just use good old fashion SQL or NoSQL, you choose.

Best Fit Use Cases

Harper DB

Operational Analytics
Data Platform & Dashboards
IoT & Industrial Systems
Monitoring & Detection
Customer Interaction Data
Gaming
(Document, streaming, and key value database use cases)

Macrometa

Graph database use cases

	HarperDB	Macrometa
DBaaS	✓	✓
FaaS	✓	✓
Performance		
On-prem latency	2 ms	No on-prem
Cloud latency	50 ms	50 ms
Read volume	120,000/sec/node	300 total
Write volume	20,000/sec/node	transactions/second
Topology		
Cloud agnostic	✓	No
On-prem	✓	No
Container friendly	✓	No
Configurable geolocation	✓	✓
Database		
Database types	Document/ Streaming/ Key Value	Document/ Dynamo/ Key Value/ Graph
ANSI-standard SQL	✓	No
Universal indexing	✓	No
ACID compliance	@node	@node
Eventual consistency	@cluster	@cluster
Non-blocking writes	✓	✓
Replication		
Configurable replication (via pub/sub)	✓	No
Can keep subset of data local	✓	No
Self-healing mesh network	✓	No
Function Development		
Natively supported languages	JavaScript & Python	JavaScript & Python
Use existing developer tools and workflows	✓	No
Start and manage background processes	✓	No
Use NPM packages	✓	No
Integration/Interfaces		
Microsoft PowerBI	✓	No
Tableau	✓	No
ODBC	✓	No
JDBC	✓	No
HTTP	✓	✓
WebSocket	✓	✓
MQTT	✓	No
Deployment		
Easily integrate into existing CI/CD pipeline	✓	No
Easily leverage GitHub actions	✓	No