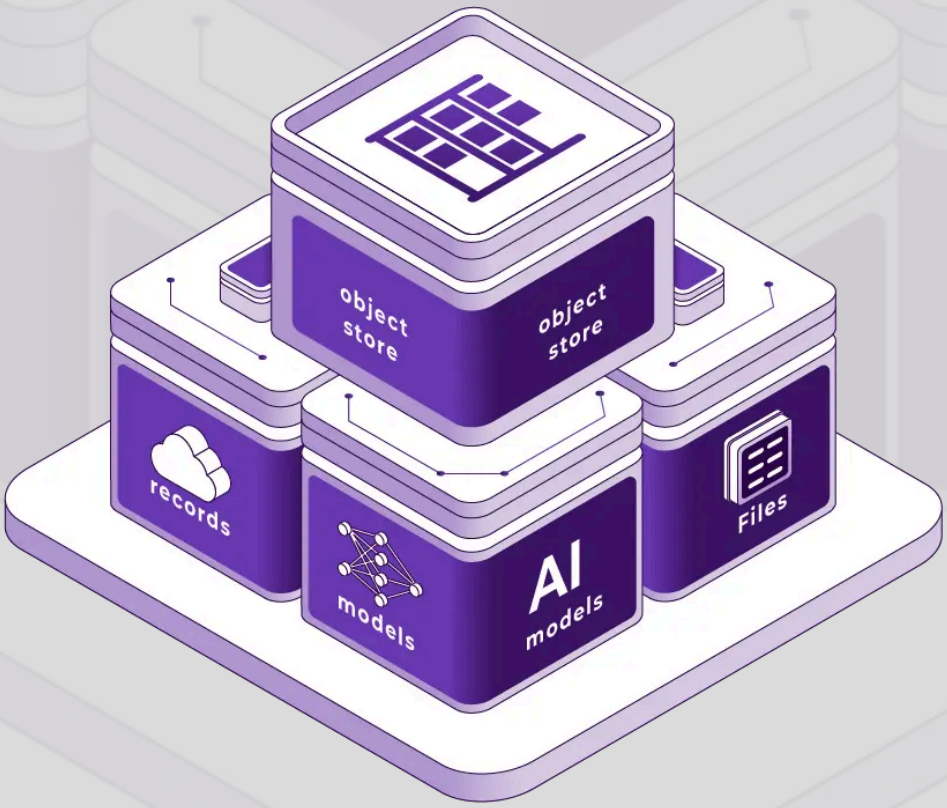
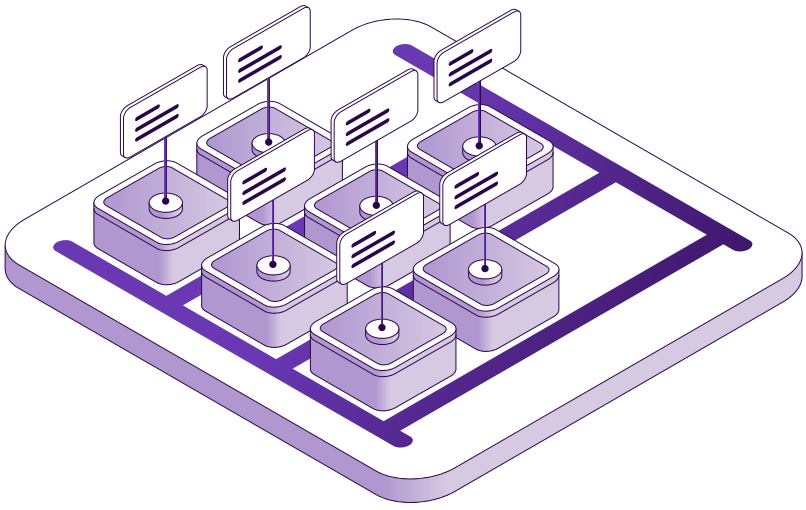
ReductStore

An Efficient Time-Series Database for IoT and Edge Computing in AI infrastructure



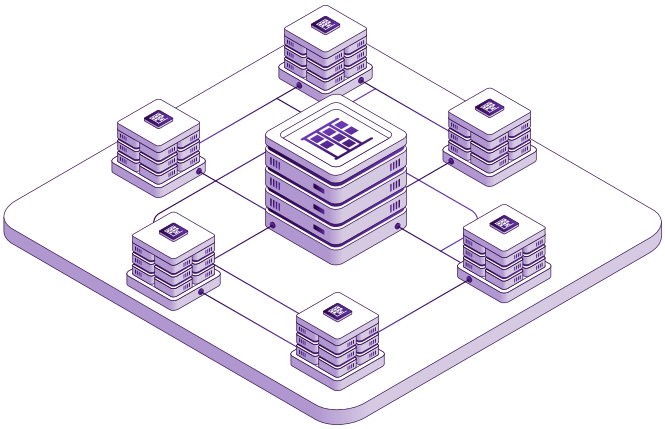
Abstract

The white paper explores various data storage solutions, focusing on their suitability for different applications, particularly in the context of time series data and unstructured data management in AI infrastructure. Traditional time series databases like InfluxDB, OpenTSDB, and TimescaleDB are highlighted for their strengths in handling structured data with high write throughput and complex queries. InfluxDB, for instance, is noted for its analytics and real-time monitoring capabilities, while OpenTSDB excels in long-term storage and analysis of massive datasets thanks to flexible querying capabilities. The aim of this paper is to explore the solution that supports unstructured data for the Internet of Things (IoT), edge computing, and Artificial Intelligence (AI) applications. An important challenge in these sorts of networks is effectively managing an increasing amount of data from many sources and diverse forms of time series data in order to meet the performance demands of applications. Time series data management in IoT is crucial for optimizing operations and has emerged as a significant area of academic and industrial research. The main objective of this work is to evaluate the current data management methods used in the IoT space and challenge the status quo, with a particular focus on time series unstructured data.



Introduction:

Every day, enormous volumes of raw data are being produced due to the continuing expansion of many applications, including the IoT, industrial monitoring, surveillance systems, and others.



No matter what kind of intelligence is used - AI, robotics, or the IoT - consumers are incredibly excited about “smart” things, at homes, in their vehicles, or even appliances. Data structured into simple and understandable ideas with AI is helping companies discover new business opportunities in the exabytes of data generated by edge devices.

In terms of industry, the IoT trend has given rise to a specific part of the IoT market called the industrial Internet of Things (IIoT) or Industry 4.0. Industry 4.0 refers to the fourth industrial revolution, characterized by the integration of advanced technologies such as AI, robotics, and the IoT into industrial processes. To get an idea of the scale, it was predicted that in 2020, Industry 4.0 and IoT would have already connected over 30 billion devices and sensors worldwide to the internet [1].

Countless sensors embedded in a wide variety of IoT devices provide massive amounts of time data at an estimated compound annual growth rate (CAGR) of 26% [2]. Data management is particularly important for performing machine learning (ML) on IoT data, and it should be done both in the cloud and at the edge to reduce latency and network traffic [3]. As a result, time series management is now subject to new obligations.

Simultaneously, 80% of the data available is unstructured, with a significant portion being generated in real-time [4]. However, only 18% of organizations are presently capable of making use of this opportunity [5].

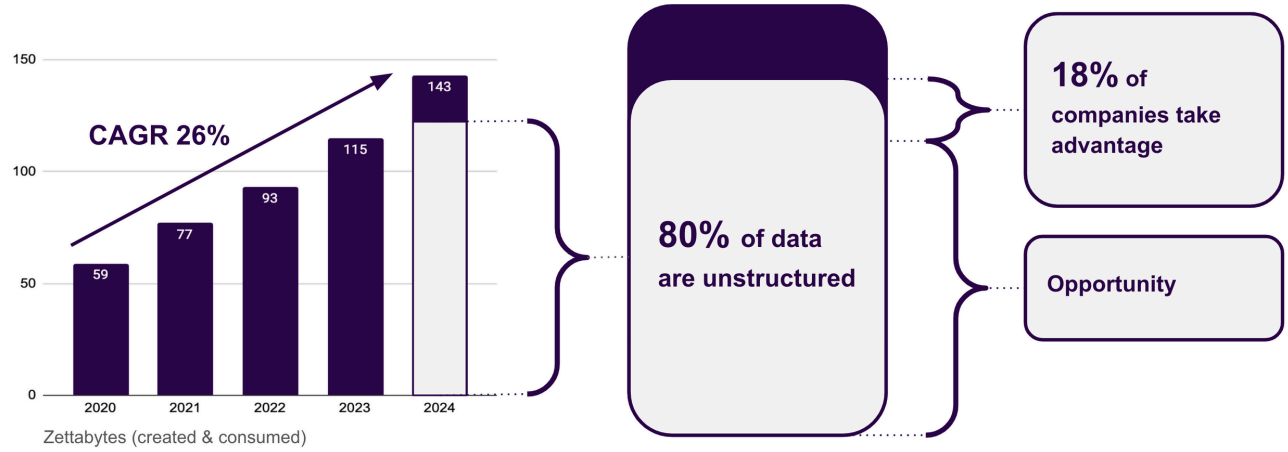
[1] Munirathinam, Sathyan. "Industry 4.0: Industrial internet of things (IIOT)." In Advances in computers, vol. 117, no. 1, pp. 129-164. Elsevier, 2020.

[2] IDC. "IDC's Global DataSphere, 2021," 2021.

[3] Ghosh, Ananda Mohon, and Katarina Grolinger. "Edge-cloud computing for Internet of Things data analytics: Embedding intelligence in the edge with deep learning." IEEE Transactions on Industrial Informatics 17, no. 3 (2020): 2191-2200.

[4] Mishra, Suyash, and Anuranjan Misra. "Structured and unstructured big data analytics." In 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), pp. 740-746. IEEE, 2017.

[5] Deloitte. "2019 Survey by Deloitte: Analytics and AI-driven enterprises thrive in the Age of With," 2019.



Data growth and organizational utilization of unstructured data

The use of IoT devices and real-time data storing and querying requires the storage of durable data, which will be queried by applications over time. These queries are often handled by time-series databases; however, cloud-based time-series storage might be prohibitively expensive as a result of its inherent complexity [6]. The proliferation of smart devices presents an opportunity to shift resilient time-series data storage and analytics to the edge of the network. This potential is made possible by the amount of computational power and memory that is accessible in edge computing.

For IoT, the edge paradigm makes more sense. It is a promising edge technology that moves storage to the edge of the network to better serve (IoT) applications in industry 4.0 that require a lot of compute. For data-intensive applications, Binary Large Objects, or Blobs, are a storage paradigm that is becoming more and more common [7]. At the same time, a comprehensive amount of control over the data is afforded to the user by the low-level, binary access mechanisms that Blob delivers. This makes it possible to optimize for individual applications, something that structured storage systems like relational databases or key-value stores can't do.

[6] Wang, Zhiqi, and Zili Shao. "TimeUnion: An Efficient Architecture with Unified Data Model for Timeseries Management Systems on Hybrid Cloud Storage." In Proceedings of the 2022 International Conference on Management of Data, pp. 1418-1432. 2022.

[7] Matri, Pierre, Alexandru Costan, Gabriel Antoniu, Jesús Montes, and María S. Pérez. "Týr: Efficient Transactional Storage for Data-Intensive Applications." PhD diss., Inria Rennes Bretagne Atlantique; Universidad Politécnica de Madrid, 2016.

Traditional DB systems:

The use of SQL-based data querying has become less efficient as the volume of data continues to increase. In particular, the management of larger databases has become a significant difficulty [8]. A growing number of businesses are relying on non-relational databases (NoSQL) to store their massive volumes of unpredictable data. Due to the following reasons NoSQL becoming increasingly popular among businesses that collect vast amounts of data:

- The primary motivation for transitioning to NoSQL databases is the requirement for efficient storage, scalability, and performance in handling large volumes of data [8].
- Data created by sophisticated apps, cloud computing, and intelligent devices can be stored and analyzed using NoSQL through a variety of interfaces [9].
- Big Data analysis with these NoSQL data models is straightforward and easy to implement, and it doesn't require complex SQL optimization techniques.
- The primary catalyst for the emergence of NoSQL databases was the availability of numerous databases that developers could utilize independently of traditional legacy systems [10].

Problems with latency and bandwidth could arise from centralized data processing. One way to address these difficulties and guarantee faster and more secure data processing is by distributing processing jobs closer to the data source. It is frequently not feasible to store massive volumes of unstructured data. Filtering and other data reduction techniques are critical to efficient long-term storage management to retain only the most relevant information.

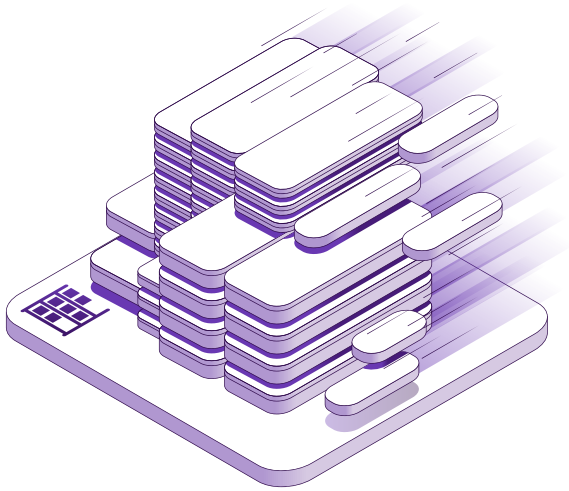
[8] Ali, Wajid, Muhammad Usman Shafique, Muhammad Arslan Majeed, and Ali Raza. "Comparison between SQL and NoSQL databases and their relationship with big data analytics." Asian Journal of Research in Computer Science 4, no. 2 (2019): 1-10.

[9] Raj, Pethuru. "A detailed analysis of nosql and newsql databases for bigdata analytics and distributed computing." In Advances in Computers, vol. 109, pp. 1-48. Elsevier, 2018.

[10] "Knowledge Base of Relational and NoSQL Database Management Systems," DB-Engines. [Online]. Available: https://db-engines.com/en/ranking. [Accessed: 29-Jul-2024].

Analysis of Existing Solutions:

InfluxDB [11] is largely acknowledged to be the most prominent time-series database that is centralized around the world [12]. A sequential series is used to store data points that have timestamps attached to them. Both fields and tags can be included in a data point's composition. Tags are optional, but fields are required for every data point. Furthermore, InfluxDB supports measurement-specific retention policies to manage data retention duration and replication frequency, allowing users to automate the downsampling and expiration of old data through the combination of retention policies and continuous queries.



An open source core provided by InfluxData is free and includes Telegraph, InfluxDB, Chronograph, and Capacitor (referred to as the TICK stack) [13]. InfluxDB is schemaless, meaning it does not require a predefined schema. However, it is not designed for traditional CRUD operations. As a time-series database, it is optimized for fast data ingestion and querying, but not for frequent updates and deletions.

OpenTSDB [14], built on top of HBase, scales well horizontally and uses tags to increase data dimensionality but provides millisecond resolution, which may not be suitable for high-frequency real-time applications. In contrast, TimescaleDB [15], an extension of PostgreSQL, breaks tables into smaller chunks, processes data in a column-oriented style, and can compress data, reducing storage and allowing individual column management.

MongoDB [16] is a flexible NoSQL database that stores data in JSON-like documents, ideal for real-time analytics and content management. It supports indexing, replication, and sharding for scalability. However, it may not perform well with high write volumes and complex transactions, and is less optimized for time-series data than dedicated solutions.

MinIO [17] is an open source object store that is compatible with Amazon S3. It is characterized by high performance and scalability, suitable for storing large amounts of unstructured data. MinIO supports erasure coding, bit-rot protection, and high throughput, but is not designed for complex data relationships or transactions.

OpenIO [18] is an open source object storage solution designed for flexibility and scalability. It can be deployed in the cloud, on-premises, or in hybrid environments, offers S3 API compatibility, and ensures data integrity through replication and erasure coding.

[11] "InfluxDB," InfluxData. [Online]. Available: https://www.influxdata.com/. [Accessed: 29-Jul-2024].

[12] Jama Mohamud, Nuh, and Mikael Söderström Broström. "Assessing Query Execution Time and Implementational Complexity in Different Databases for Time Series Data." (2024).

[13] Naqvi, Syeda Noor Zehra, Sofia Yfantidou, and Esteban Zimányi. "Time series databases and influxdb." Studienarbeit, Université Libre de Bruxelles 12 (2017): 1-44.

[14] OpenTSDB. [Online]. Available: http://opentsdb.net/. [Accessed: 29-Jul-2024].

[15] TimescaleDB. [Online]. Available: https://www.timescale.com/. [Accessed: 29-Jul-2024].

[16] MongoDB. [Online]. Available: https://www.mongodb.com/. [Accessed: 29-Jul-2024].

[17] MinIO. [Online]. Available: https://min.io/. [Accessed: 29-Jul-2024].

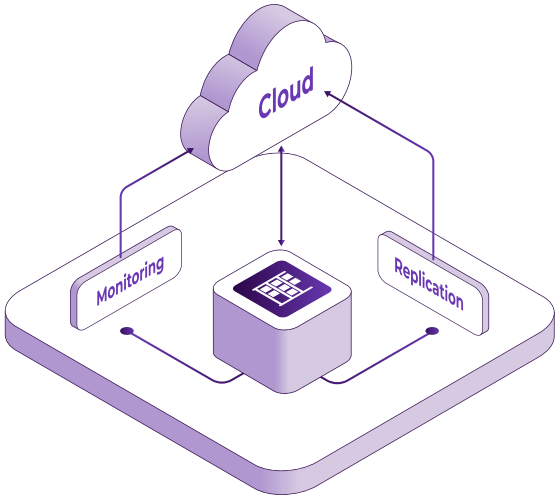
[18] OpenIO. [Online]. Available: https://github.com/open-io/. [Accessed: 29-Jul-2024].

Table 1: Comparison of Traditional System

System	Data Structure	Query Language	Performance	Application
InfluxDB	Structured time series data with tags and fields.	InfluxQL, a SQL-like query language, and Flux, a data scripting language.	Good in high write throughput and complex queries	Best for systems that need analytics and real-time monitoring with structured data.
OpenTSDB	Structured time series data with metrics and tags, based on HBase.	Uses a query language specific to OpenTSDB.	Scales well horizontally, making it ideal for very large datasets.	Suitable for long-term storage and analysis of massive time series datasets.
Timescale DB	Based on the relational data model of PostgreSQL.	Uses SQL for querying, leveraging the full capabilities of PostgreSQL.	Optimized for high ingestion rates and complex queries through hyper tables and chunking.	Ideal for applications needing relational data features along with time series data management.
MongoDB	Uses a flexible, document-oriented data model.	Uses MongoDB Query Language (MQL).	Can handle large datasets and high throughput with appropriate indexing.	Fits flexible schema and different data storage demands, including time series data.
MinIO	Stores data as objects in buckets, similar to AWS S3.	Does not have a native query language but supports APIs for data access (like AWS S3).	Optimized for large-scale object storage with high availability and redundancy.	Ideal for applications needing scalable and resilient object storage.
OpenIO	Stores data as objects with a flat structure.	Provides an API for data access but no native query language.	Scales horizontally and supports high availability and redundancy.	Suitable for cloud storage, big data, and backup solutions.

Challenging the Traditional Systems:

In practical scenarios, time series databases are optimized for structured data with well-defined structures, allowing for efficient querying and analysis. On the other hand, object storage systems are designed to handle unstructured data, offering scalability and redundancy, but lacking the specialized querying capabilities required for time series analysis. This separation means that applications often have to choose between the efficiency of structured time series databases and the flexibility of object storage, without a unified solution that can handle both in tandem.



ReductStore [19] is a specialized system created to effectively handle and store unstructured data that is organized based on time. The system is composed of buckets, entries, blocks, and records. Buckets are receptacles that store data entries and possess configurable attributes such as maximum block size, maximum record count per block, and quota types to govern storage restrictions. ReductStore is designed to manage massive volumes of "blob" data, which is unstructured data. Its adaptability and ease of integration with other applications are due to its support for an HTTP API. Applications in edge computing that are performance-sensitive greatly benefit from ReductStore's design, which prioritizes efficient storage and retrieval of binary data.

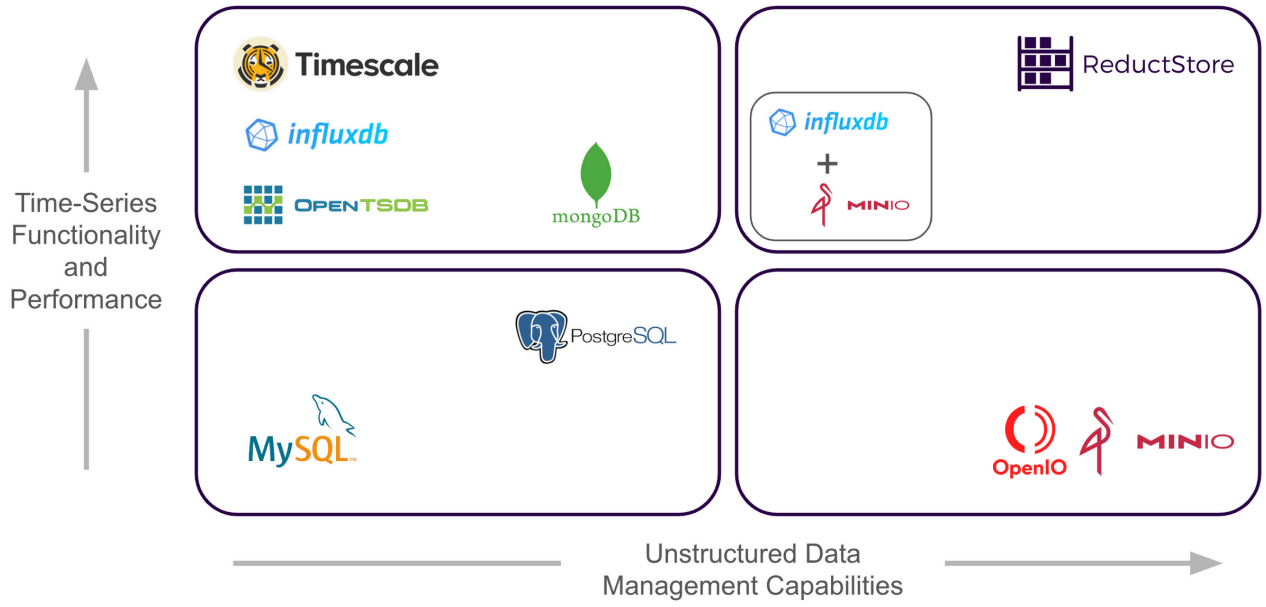
The ReductStore team focuses on building a time series object store for managing unstructured data, with the goal of improving the use of artificial intelligence and edge computing services, as well as overall performance. ReductStore's contributors come primarily from the IIoT industry and have built a storage management solution from the ground up.

They give customers and organizations access to a wide variety of opportunities, such as increased efficiency and a reduction in the amount of time needed for queries of unstructured time series data. When compared to TimescaleDB and InfluxDB, ReductStore stands out due to significantly better performance for data sizes of 10kB and above. At the same time, ReductStore's time series API and batching capabilities makes it incomparable to other S3 storage options, such as MinIO.

Table 2: Comparative Analysis of ReductStore with Other Solutions [19]

	VS TimeScale		VS MongoDB		VS MinIO	
Record Size	Read Speed (%)	Write Speed (%)	Read Speed (%)	Write Speed (%)	Read Speed (%)	Write Speed (%)
10 MB	+850%	+1300%	+65%	+158%	+10%	+361%
1 MB	+855%	+1075%	+112%	+137%	+79%	+486%
100 KB	+217%	+205%	+198%	+155%	+677%	+258%

ReductStore efficiently manages large datasets with features that optimize both time-series functionality and object storage management capabilities.



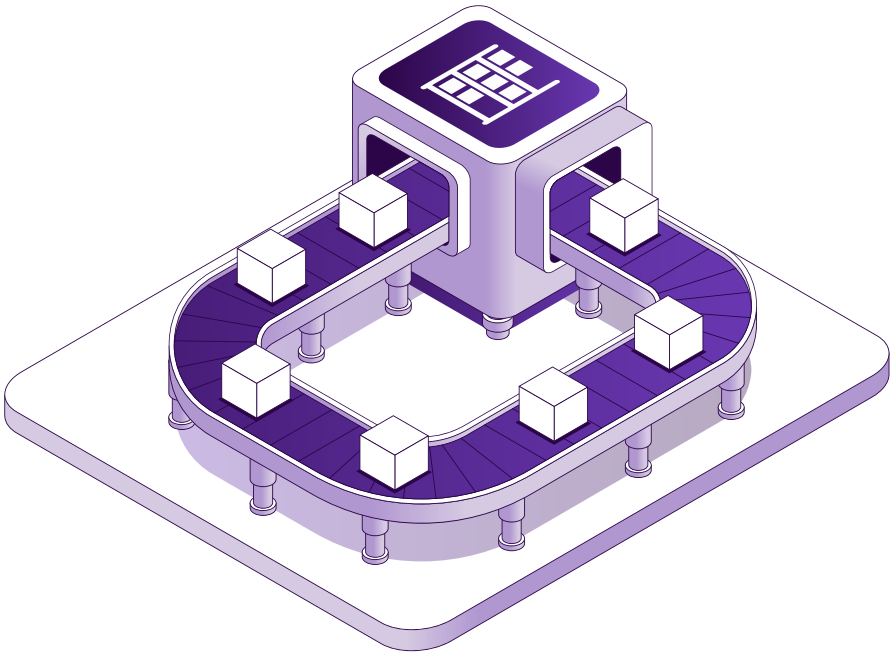
Comparison of database capabilities: Time-Series functionality and performance vs. unstructured data management

[19] ReductStore. [Online]. Available: https://www.reduct.store/. [Accessed: 29-Jul-2024].

Specific features for edge computing:

ReductStore's real-time, first-in, first-out (FIFO) quota system ensures that disk space is optimally managed, preventing shortages that could disrupt operations on edge devices. For example, a strict FIFO quota can be applied to high-frequency vibration data from industrial machinery, image data from computer vision applications, or any complex IoT application where each record is inherently large.

In addition, each record can be associated with labels - typical in AI applications - which are key-value pairs and can hold meta-information about the record. These labels can also be used for filtering and replication. When it comes to network challenges, it also offers a feature to minimize network overhead, especially in high latency environments, by batching data retrieval operations.



Cost Analysis:

> **« ReductStore is a vital part of our infrastructure. It handles terabytes of unstructured data in a production environment. »**
>
> *Michael Welsh, Founder at Metric Space UG*

To illustrate the benefits of ReductStore, let's consider an organization that processes and transfers 50 terabytes of data per month, with each blob averaging 10KB in size. If 1TB consists of 100 million 10KB blobs, we can calculate the transfer time for those 100 million blobs using the following benchmark [20]:

Blob Size	Operation	MongoDB, blob/s	ReductStore, blob/s	ReductStore, %
10 KB	Write	529	1531	+190%
10 KB	Read	379	1303	+244%

Write Operations:

MongoDB: $\frac{\text{100 million blobs}}{\text{529 blobs / sec}} = 53 \text{ hours}$

ReductStore: $\frac{\text{100 million blobs}}{\text{1531 blobs / sec}} = 18 \text{ hours}$

Write Operations:

MongoDB: $\frac{\text{100 million blobs}}{\text{379 blobs / sec}} = 73 \text{ hours}$

ReductStore: $\frac{\text{100 million blobs}}{\text{1303 blobs / sec}} = 21 \text{ hours}$

Time savings per terabyte:

- Write operation savings: 35 hours (53 hours for MongoDB - 18 hours for ReductStore)
- Read operation savings: 52 hours (73 hours for MongoDB - 21 hours for ReductStore)

[20] "Alternative to MongoDB for Blob Data" [Online]. Available: https://www.reduct.store/blog/comparisons/iot/reductstore-vs-mongodb/. [Accessed: 29-Jul-2024].

Cloud providers typically charge for data transfer based on the volume of data transferred. However, there are additional costs associated with the time it takes to transfer data, such as resource usage and CPU time. Long data transfer times can also result in suboptimal application performance because the system cannot carry out other critical actions during this time.

To provide a rough estimate of cost savings, we'll assume certain parameters for illustrative purposes. While it's difficult to determine an exact cost, we can estimate the cost range for the sake of this example using a cost of $10 per hour. We estimated a total time savings of 87 hours per terabyte (35 hours for writes + 52 hours for reads). Using an estimated hourly transfer cost of $10/hour, the calculations are as follows:

- Cost savings per terabyte: 87 hours × $10 / hour = $870
- Total monthly savings: 50 terabytes × $870 / terabyte=$43,500
- Total annual savings: $43,000 per month × 12 months = $522,000

This simple analysis shows that the time it takes to transfer data has a significant impact on overall costs due to resource consumption. By optimizing data transfer times with solutions like ReductStore, companies can realize significant cost savings. While precise cost estimates are difficult due to variable factors, this example illustrates potential cost savings and underscores the importance of efficient data management for organizations dealing with large volumes of unstructured data.

Conclusion:

After careful consideration, we have determined that the systems that we have selected are representative of the most advanced unstructured storage solutions for edge computing and industry 4.0. ReductStore is well-suited for scenarios involving edge computing and the IoT that require the effective storage and rapid retrieval of massive quantities of binary data because of its emphasis on unstructured data, efficient batching, and FIFO quota management. ReductStore offers a unique solution tailored for unstructured time series data, making it a strong contender for edge computing applications in AI infrastructures where managing large blobs efficiently is critical. Its simplicity, performance optimizations, and focus on unstructured data along with labels set it apart from more traditional time series databases like InfluxDB, OpenTSDB, and TimescaleDB, as well as object storage solutions like MinIO and OpenIO.

For applications requiring efficient handling of binary data with straightforward integration and high write performance, ReductStore is an excellent choice. However, for scenarios needing detailed analytics and real-time monitoring of structured data, traditional time series databases remain highly effective solutions. Similarly, for large-scale object storage needs, Minio and OpenIO offer robust and scalable options. Finally, whilst conventional time series databases are still great for analytics on structured data and real-time monitoring, ReductStore provides a strong substitute for unstructured data in edge computing. As companies face new challenges in efficiently managing and analyzing massive amounts of unstructured data, they are increasingly turning to NoSQL databases.

🔗 LinkedIn - ReductStore

💬 Join us on Discord

✉ info@reduct.store

🌐 www.reduct.store