Apache IoTDB

A Time Series Database for IoT Applications

Yuan Tian
Apache IoTDB PMC Member

SIEMENS

Outline

A

Introduction

B

Our Solutions

C

Competitive Advantages

D

Use Cases

SIEMENS

Introduction

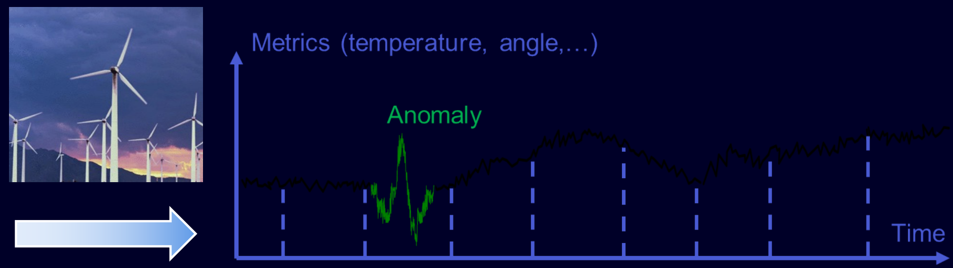
What Is TSDB and Why Is It Important?

SIEMENS

What is Time Series Data?

- Time series data or time series is a sequence of data points recorded over time intervals, such as stock prices, sensor data, or weather data.

- In the Industrial Internet of Things (IIoT), massive time series data generated by machines and sensors accounts for 80% of all data.



Profile of Time Series Data

As a global leader and pioneer of Industrial IoT, General Electric has emphasized the importance of Time Series Date early in 2012:

The Rise of Industrial Big Data

Leveraging large time-series data sets to drive innovation, competitiveness and growth—capitalizing on the big data opportunity

Source: GE (2012). *The Rise of Industrial Big Data.*

New Demands for Data Management in the IoT

In 2019, the development of database for Industrial IoT was discussed in *"The Seattle Report on Database Research"*, written by 33 top-level scholars in the area of database researching:

> Industrial Internet-of-Things (IoT), focusing on domains such as manufacturing, retail, and healthcare greatly accelerated in the last five years, aided by versatile connectivity, cloud data services, and data analytics infrastructure. Its requirements have further stress-tested our ability to do fast data ingestion and quickly discover insights with minimal delay for real-time scenarios such as monitoring. Their effectiveness also depends on efficient data processing at the edge, including data filtering, sampling, and aggregation.

The Seattle Report on Database Research

Daniel Abadi, Anastasia Ailamaki, David Andersen, Peter Bailis, Magdalena Balazinska, Philip Bernstein, Peter Boncz, Surajit Chaudhuri, Alvin Cheung, AnHai Doan, Luna Dong, Michael J. Franklin, Juliana Freire, Alon Halevy, Joseph M. Hellerstein, Stratos Idreos, Donald Kossmann, Tim Kraska, Sailesh Krishnamurthy, Volker Markl, Sergey Melnik, Tova Milo, C. Mohan, Thomas Neumann, Beng Chin Ooi, Fatma Ozcan, Jignesh Patel, Andrew Pavlo, Raluca Popa, Raghu Ramakrishnan, Christopher Ré, Michael Stonebraker and Dan Suciu

Industrial Internet-of-Things (IoT), focusing on domains such as manufacturing, retail, and health-care greatly accelerated in the last five years, aided by versatile connectivity, cloud data services, and data analytics infrastructure. Its requirements have further stress-tested our ability to do fast data in-gestion and quickly discover insights with minimal delay for real-time scenarios such as monitoring. Their effectiveness also depends on efficient data processing at the edge, including data filtering, sam-pling, and aggregation.

	Scenarios	Performance Requirements	Data Storage	Data Processing	Data Application
New Demands	Factory Deployment Device Deployment	Tens of Millions Measure Points and more..	100% Storage	Edge-to-Cloud Synchronization	Computing/ Analyzing on Industrial Demands
Traditional					

Benefiting Industrial Enterprises with Time-series data

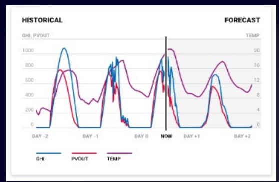
Power Industry

DCS System

Live data only

Live data + historical data

Power Generation



New Energy Power Generation

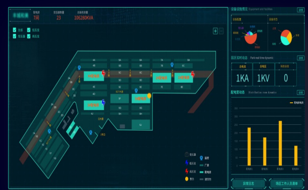
Live data + historical data

Power generation forecast

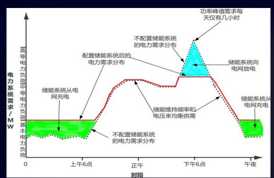
Power Transmission

Substation

Multiple-node data monitoring and analysis for failure root cause analysis



Power Distribution



Demand side management

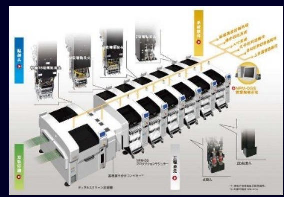
Demand analysis, demand control, peak shaving, improvement of grid dispatching

Manufacturing

Modelling

Manage massive result data and improve work efficiency

Product Design



Discrete (SMT) Manufacturing

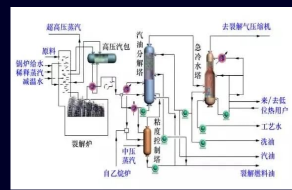
Monitor and analyze multiple measuring point data for collaborative analysis, reducing manual rechecking work

Process Control

Process Chemical Industry

Live data + historical data

Control process parameters to ensure production safety



O&M

Engineering Machinery

Anomaly detection and preventive intervention

SIEMENS

Pain Points of Data Management in the IoT

Huge Amount of Measurements

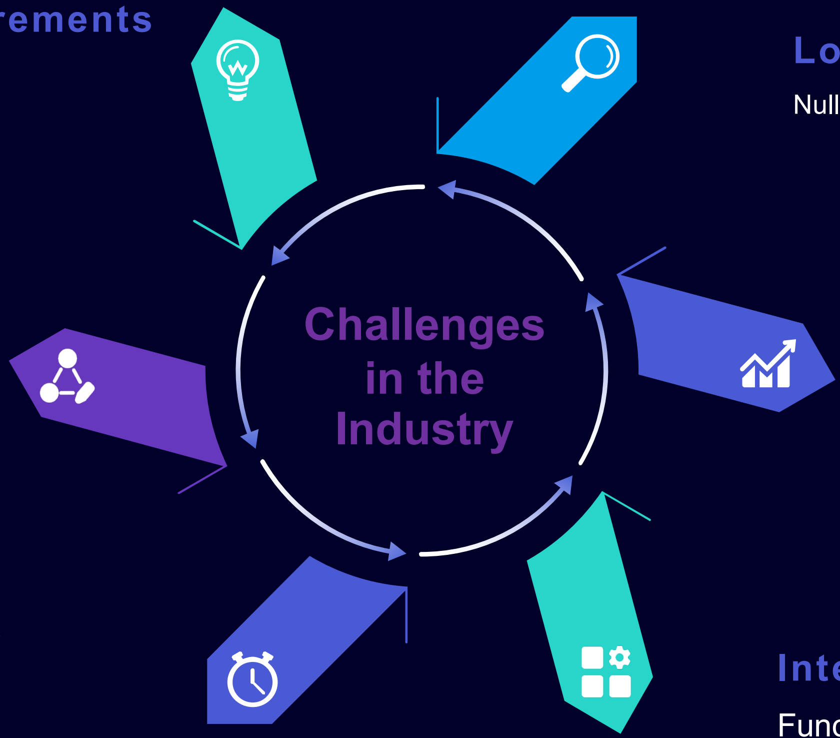
e.g. **300,000 sensors/tags** in a single generation unit

Low Data Quality

Null value, lost timestamp, etc.

High Sampling Frequency

e.g. **25,000Hz** in large power plants



Embedded-Edge-Cloud

Data management through embedded-devices, edge gateways and central server (cloud)

Unordered Data

Over **50%** data are out-of-ordered data due to bad network

Integration of IT & OT

Functionalities of data analytics for OT

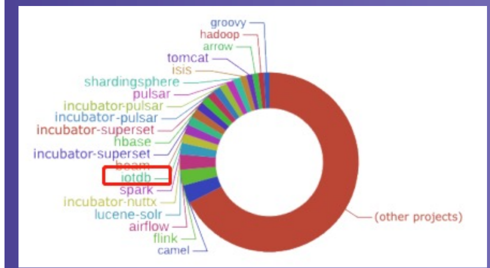
SIEMENS

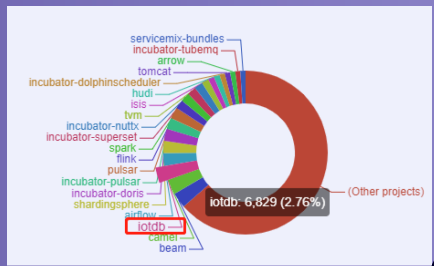
Our solutions

The Booming Open Source Community of TLP Apache IoTDB

IoTDB in 2021: Rank 7 by commits among all the 351 Apache Projects, surpassing Hadoop/ HBase, next to Spark

IoTDB in 2022: Rank 3 by commits among all the 364 Apache Projects.





Super Healthy Project with other TLPs like Apache Flink/ Hadoop/ HBase
(7.4% projects were evaluated Super Healthy)

Source: Github, Apache Annual Report 2021 & 2022

The international TSDB open source community with contributors distributed in China, the US, Germany, the UK, Australia, etc.

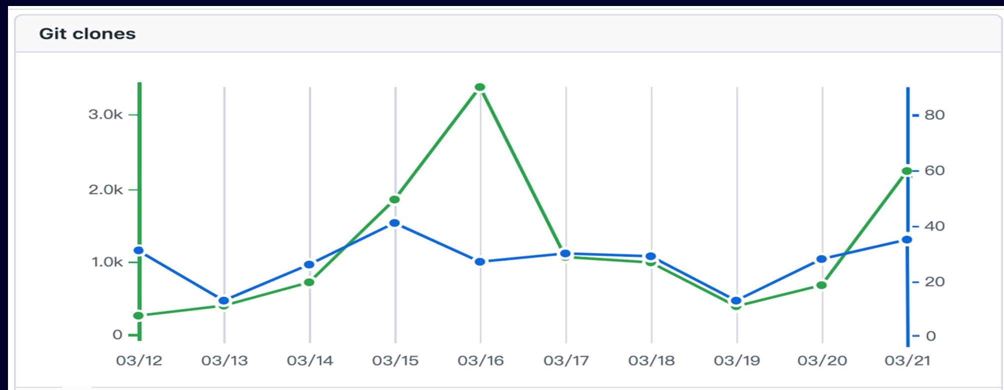
240+
Contributors

3,000+
Max. daily downloads

30,000+
Monthly source code downloads

3,000+
Stars

800+
Forks

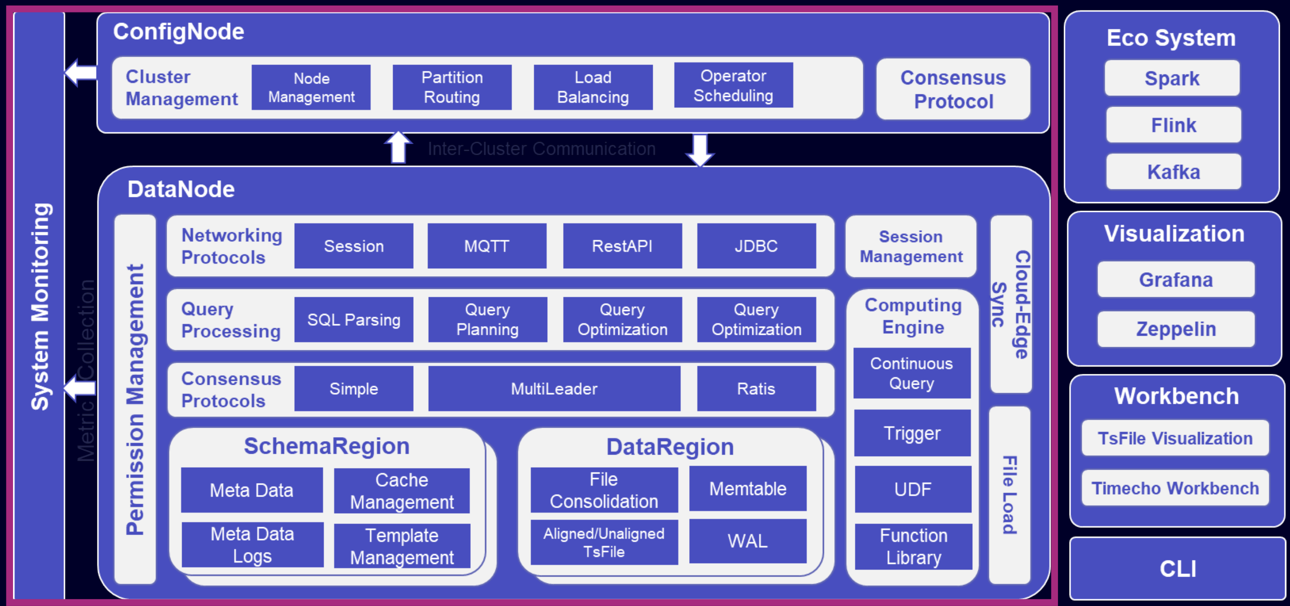


SIEMENS

Efficient Time Series Data Management

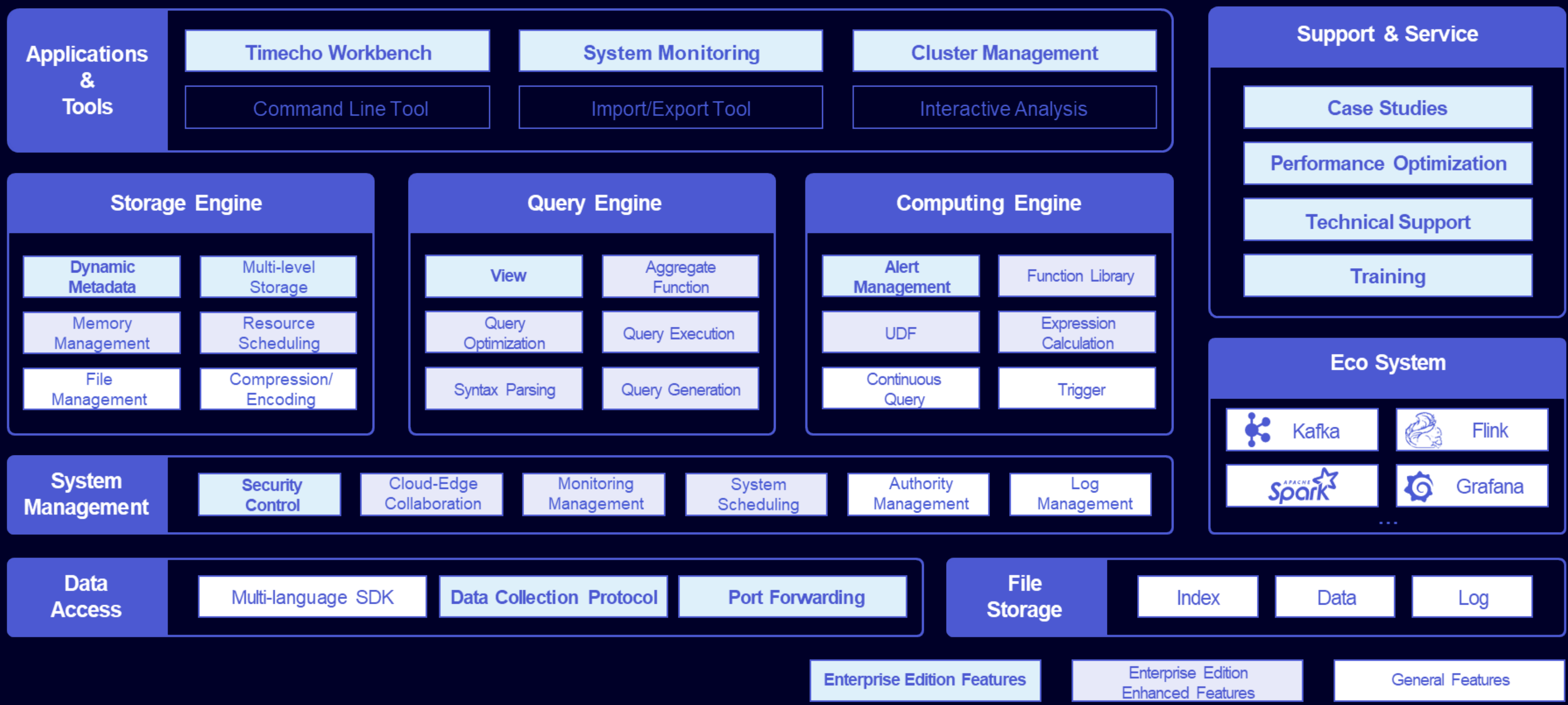
Apache IoTDB is an IoT-native data management system designed for time-series data, which provides users specific services for data collection, storage and analysis. It was designed for resolving the pain points of the typical IoT/ IIoT use cases, including massive data generation, high frequency sampling, out-of-order data, high costs of storage and O&M, low computational power of IoT devices, etc.

Homepage: https://iotdb.apache.org/
GitHub: https://github.com/apache/iotdb



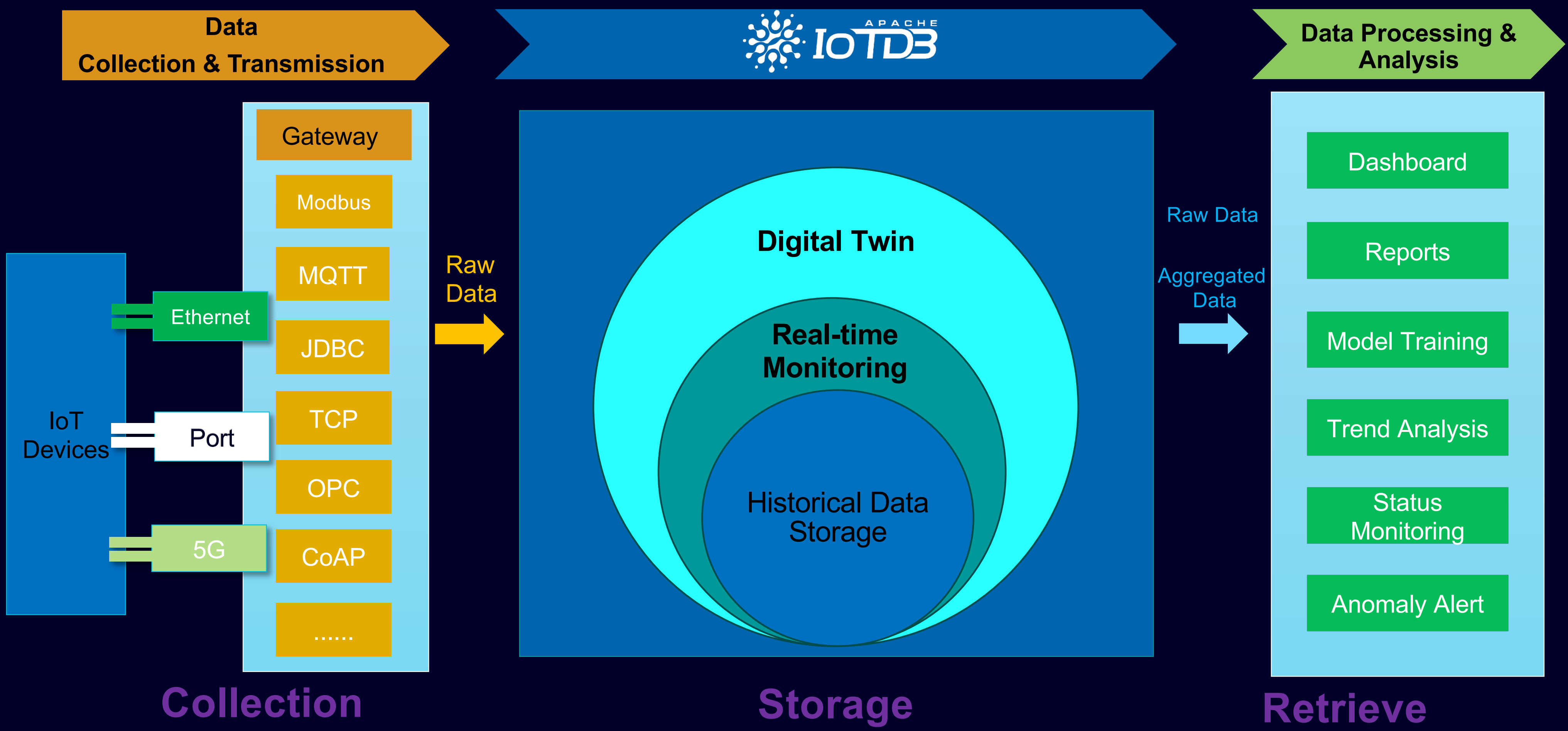
Architecture of Apache IoTDB V1.0

Apache IoTDB Architecture and Features Overview



SIEMENS

Efficient Time Series Data Management

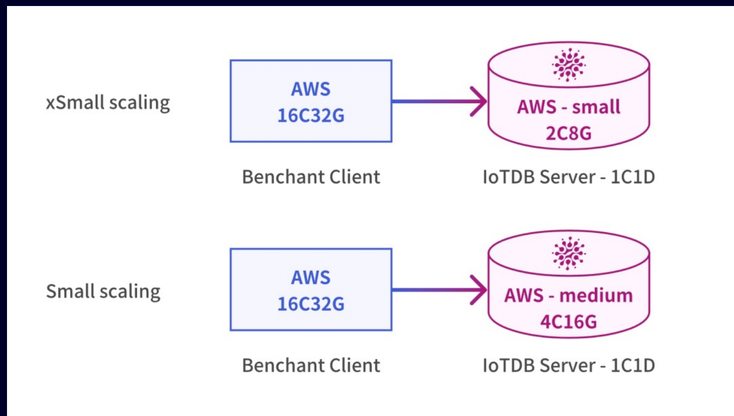


SIEMENS

BenchAnt Ranking

BenchAnt[1]

- Located in Germany, it is a third-party testing organization specializing in cloud facility and database performance evaluation.
- The evaluation criteria for the time series database rankings are based on the DevOps scenario of the TSBS[2] benchmark suite, tested in a unified AWS environment
- Tested under two hardware resource configurations



[1] https://benchant.com/
[2] https://github.com/timescale/tsbs

BenchAnt Ranking

Write Throughput in small env

- 3,636,312 ops
- 81% better than VM
- 41% better than QuestDB
- 6.8 times of InfluxDB

Storage spcae in small env

- 2 GB

RANK	DATABASE	CLOUD	WRITE THROUGHPUT [ops]	STORAGE CONSUMPTION [GiB]	*READ THROUGHPUT* [ops]	READ LATENCY [ms]	MONTHLY COSTS [$]	OPERATIONS PER COST [ops/$]
1	Apache IoTDB v1.2.1 OpenSource tuned small	AWS medium	3,636,312	2	11,497	3	178	64.59
2	VictoriaMetrics v1.7.90 OpenSource vanilla small	AWS medium	1,987,341	2	8,446	6	178	47.55
3	Apache IoTDB v1.2.1 OpenSource vanilla small	AWS medium	3,148,220	2	5,694	2	178	31.99
4	InfluxDB v1.8.10 OpenSource vanilla small	AWS medium	529,220	3	2,197	45	178	12.37
5	QuestDB v6.5.0 OpenSource vanilla small	AWS medium	2,553,554	33	715	114	178	4.03

SIEMENS

BenchAnt Ranking

QPS in small env

- 11,497 ops
- 36% better than VM
- 5 times of InfluxDB
- 16 times of QuestDB

Query Latency in small env

- 2 ~ 3 ms, much lower than others

RANK	DATABASE	CLOUD	WRITE THROUGHPUT [ops]	STORAGE CONSUMPTION [GiB]	*READ THROUGHPUT* [ops] ▼	READ LATENCY [ms]	MONTHLY COSTS [$]	OPERATIONS PER COST [ops/$]
1	Apache IoTDB v1.2.1 OpenSource tuned small	AWS medium	3,636,312	2	11,497	3	178	64.59
2	VictoriaMetrics v1.7.90 OpenSource vanilla small	AWS medium	1,987,341	2	8,446	6	178	47.55
3	Apache IoTDB v1.2.1 OpenSource vanilla small	AWS medium	3,148,220	2	5,694	2	178	31.99
4	InfluxDB v1.8.10 OpenSource vanilla small	AWS medium	529,220	3	2,197	45	178	12.37
5	QuestDB v6.5.0 OpenSource vanilla small	AWS medium	2,553,554	33	715	114	178	4.03

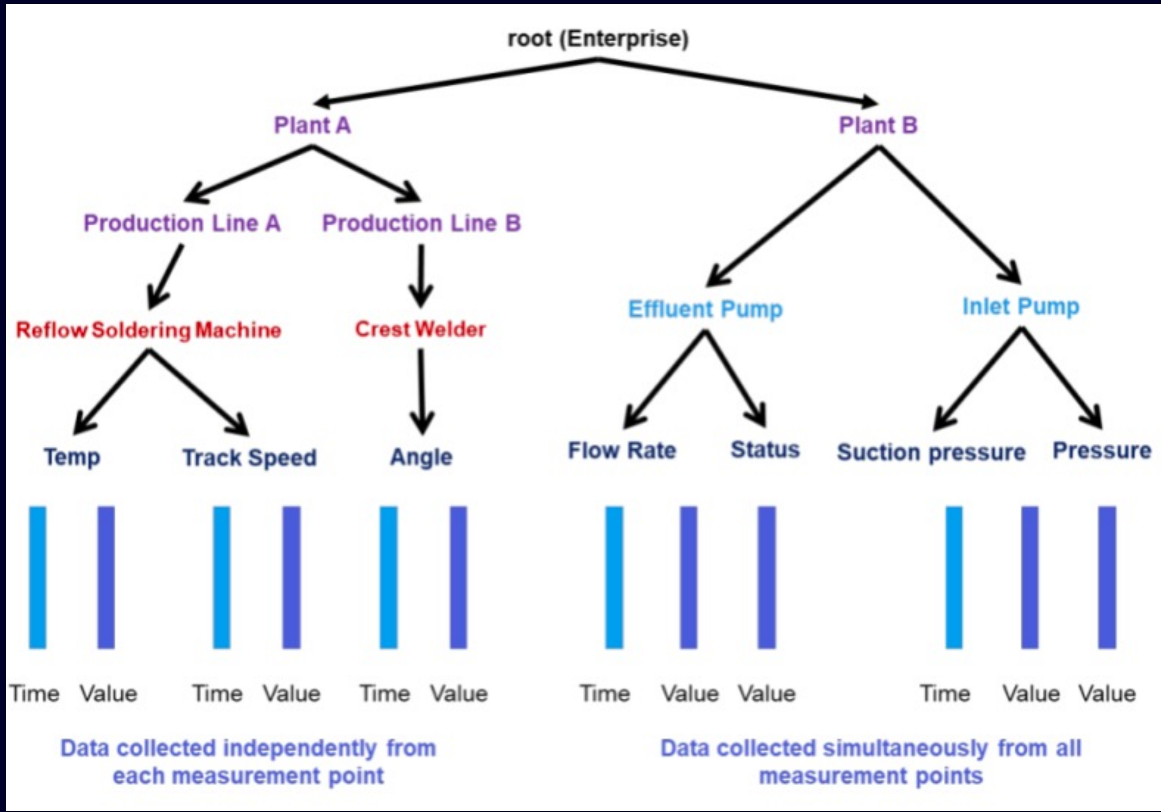
SIEMENS

Competitive Advantages

SIEMENS

Features: IoT Data Model

- **Data Model for the industry:**
  - Flexible Tree-like hierarchic structure
- **Device Templates:**
  - Supports templated management
- **Attribute grouping:**
  - Supports sequences alignment
- **Static data:**
  - Supports sequence tagging and attribute management
- **Dynamic Model:**
  - Supports automatic identification and generation of metadata

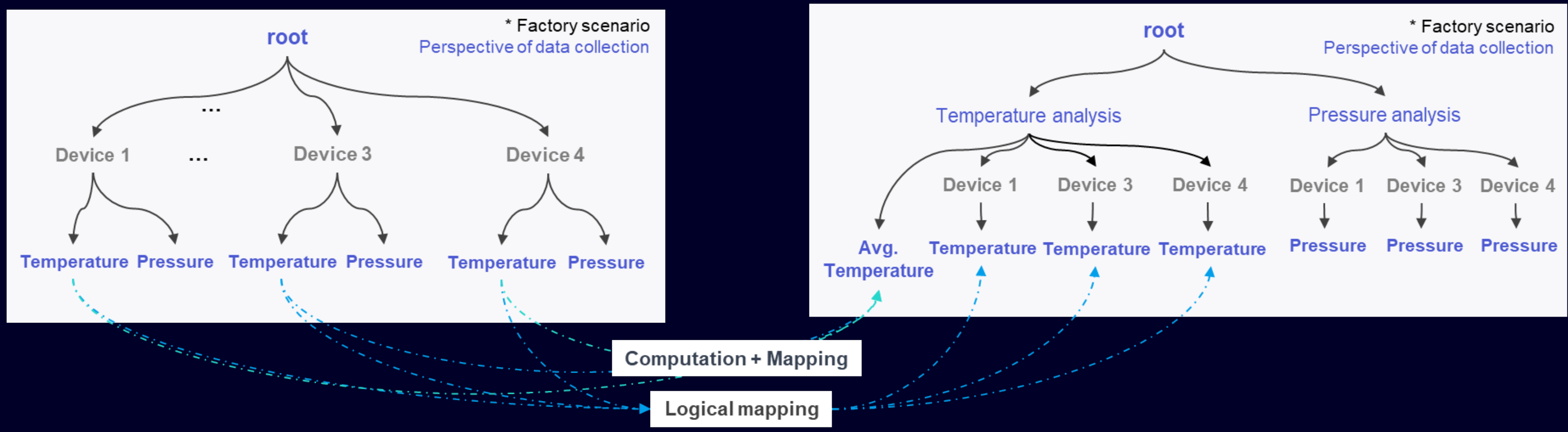


SIEMENS

Features: Analytical Views

Raw time series: persistent storage
(Management based on device location)

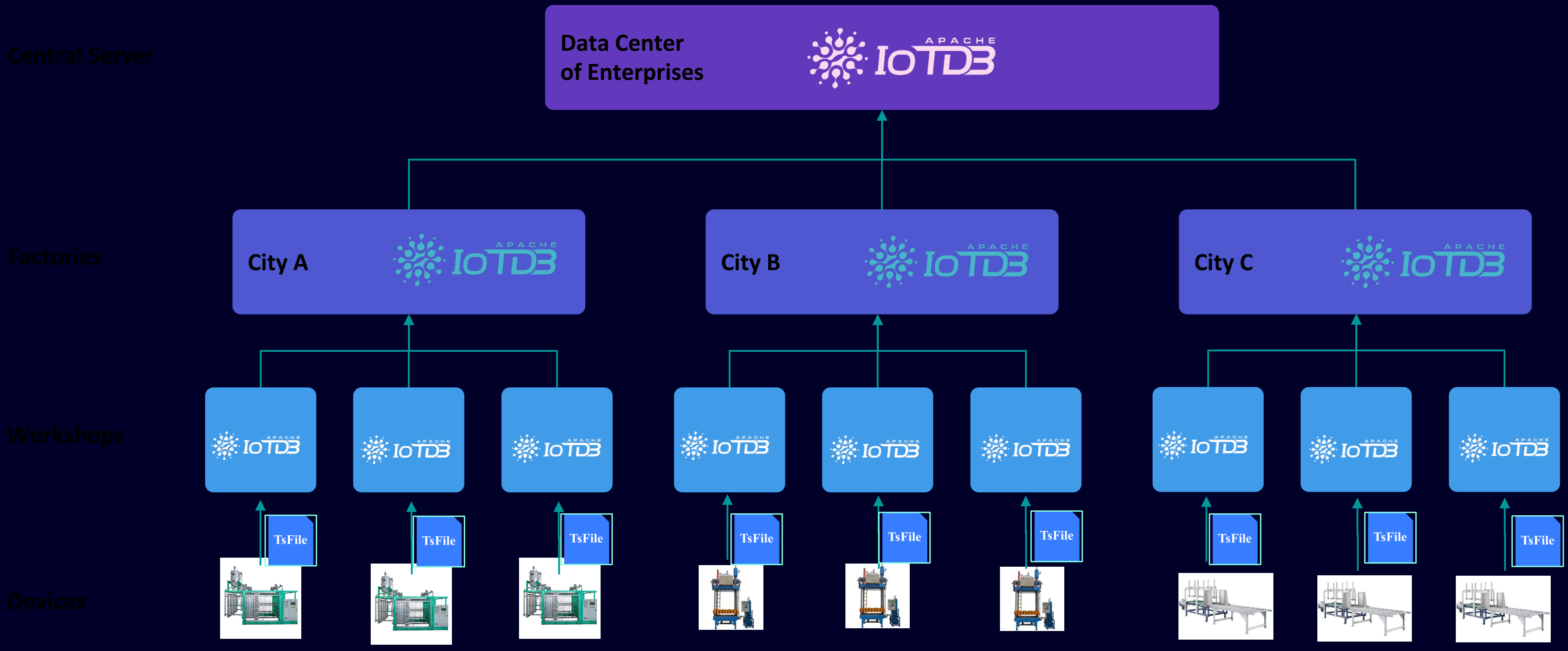
Views: individualized data structure
(Management based on analytical applications)



- **View separation**: data acquisition teams focus solely on data collection, while business teams are relieved from the need to understand device coding and business logic, resulting in reduced learning costs.
- **Enhanced Computation**: supports logical mapping and processing operations, providing enhanced computational capabilities for convenient business usage
- **Controllable Permissions**: supports partitioning business permissions based on views, ensuring controlled access to data and enhancing data security

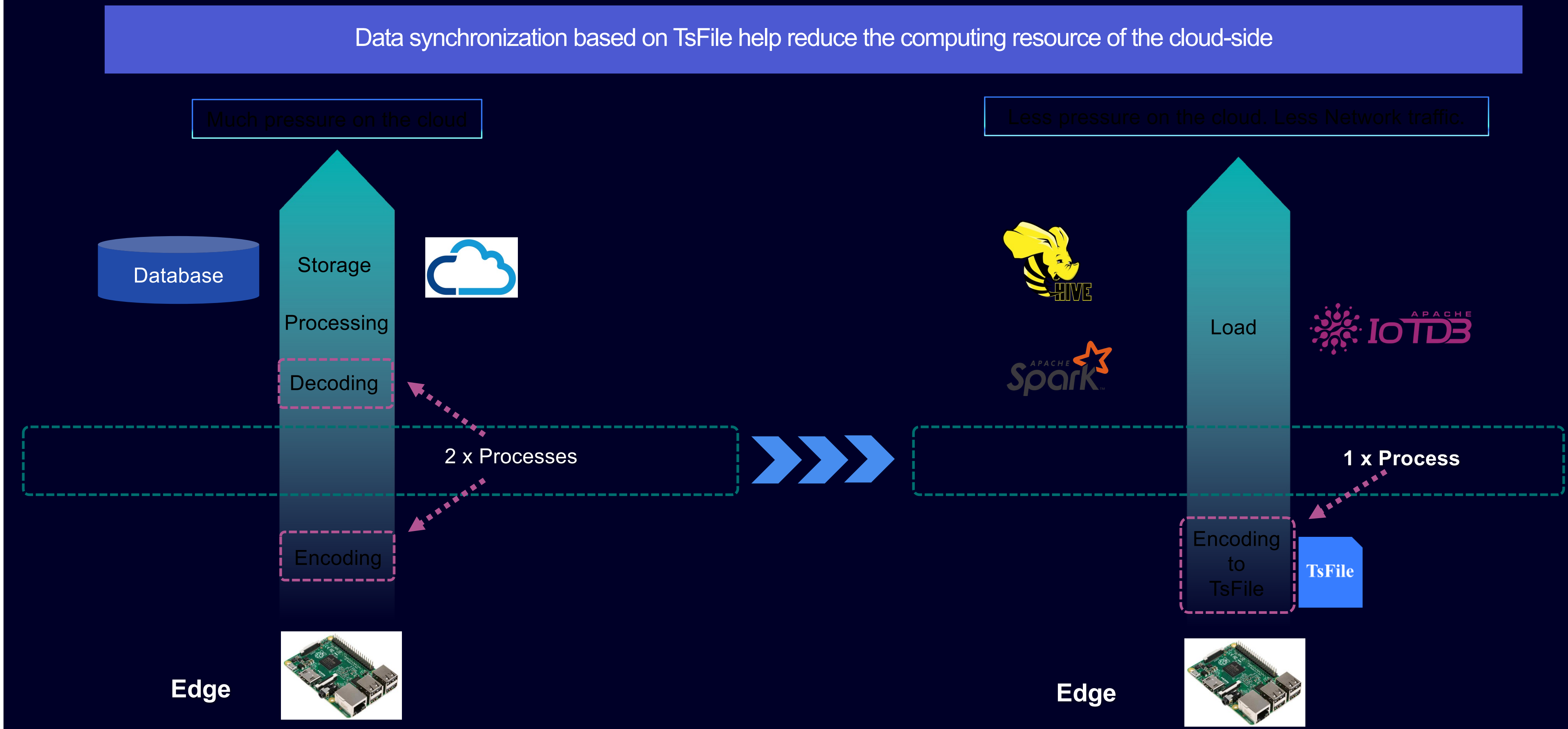
SIEMENS

Features: Cross Embedded-Edge-Cloud Deployment & Data Sync



SIEMENS

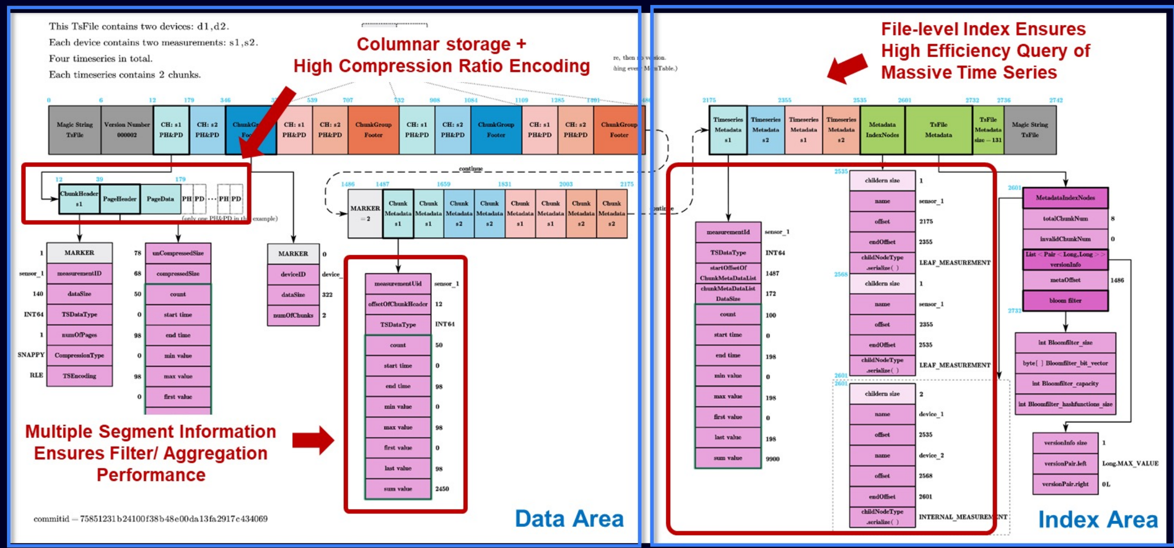
Features: Advantages in Data-Sync with “TsFile”



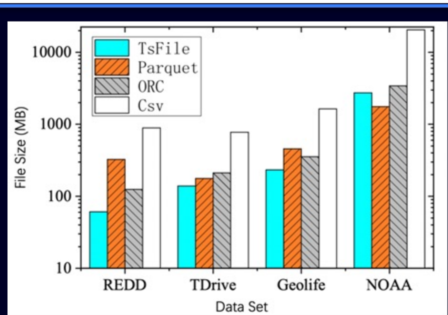
SIEMENS

Features: Columnar File Format “TsFile”

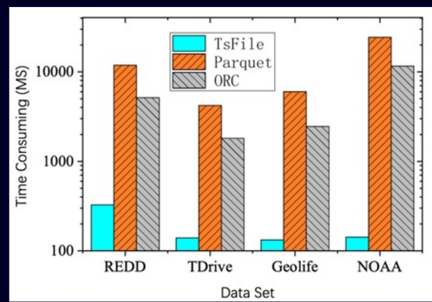
- **TsFile** is a column-oriented storage file format optimized for time series data, similar to Parquet and ORC.
- TsFile reduces the hardware resources required for data storage and optimizes the performance of data query.
- The Embedded-Edge-Cloud data synchronization is simplified with TsFile



Comparison	TsFile	Parquet	ORC
Data Model (Time Series Semantics)	Yes	No	No
Encoding of Time Series	Optimized	Normal Encoding	Normal Encoding
Device-level Index	Yes	No	No
Storage	Without Redundancy	Sequence ID Redundancy	Sequence ID Redundancy
Query Speed	Fast	Slow	Medium



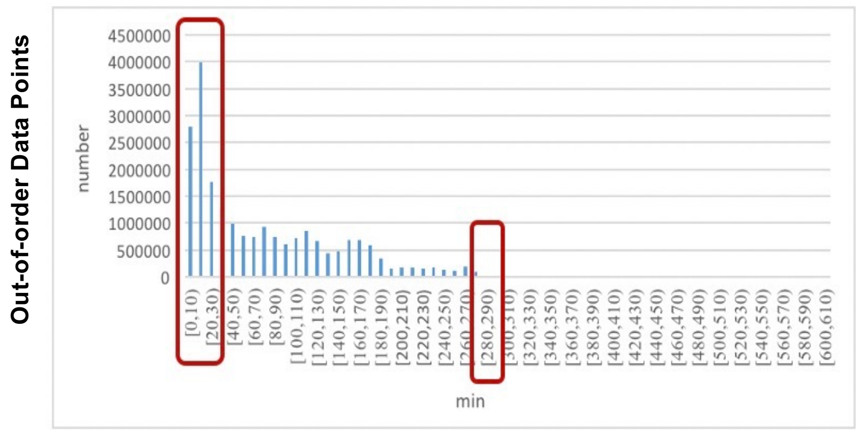
File Size in Comparison



Query Speed in Comparison

SIEMENS

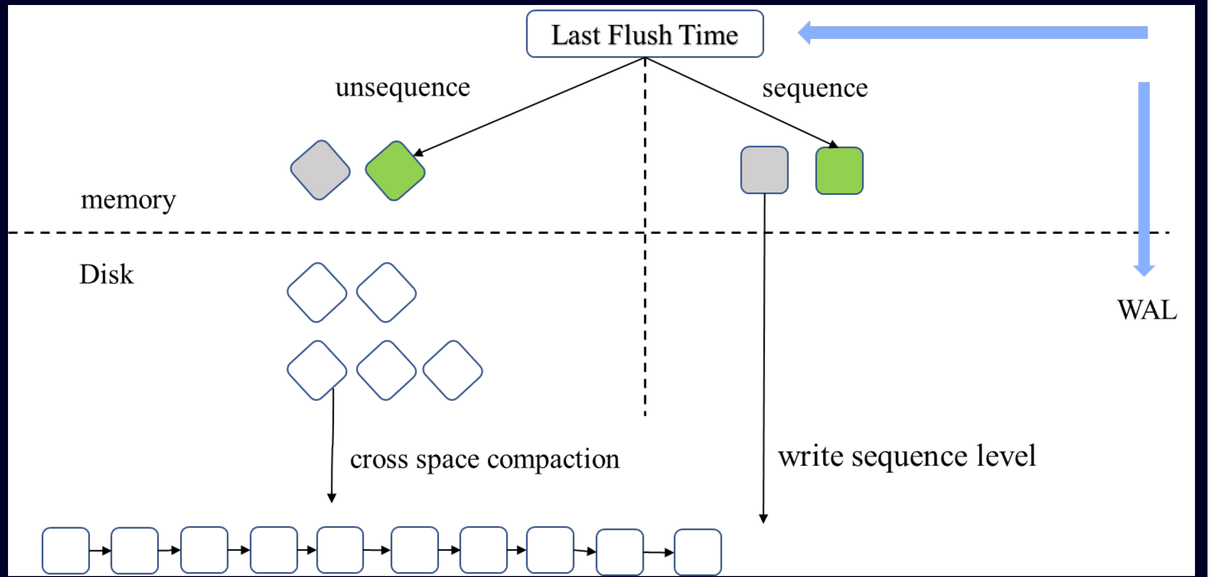
Features: Out-of-Order Data Processing



Qinghai Big Data Platform: Disordered Data in Jan 2018 (50%)

Out-of-Order (Out-of-sequence) Data:

- 50% data is out-of-order
- **Out-of-order data arrival latency ranges from 0 to 300 mins**
- Out-of-order data **with arrival latency in 30 mins** covers about 90% proportion

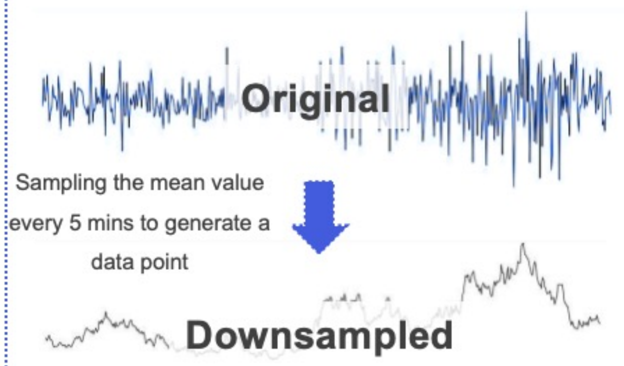


Apache IoTDB supports Out-of-Order Data Writing-in

SIEMENS

Features: Various Time-Series Querying Capability

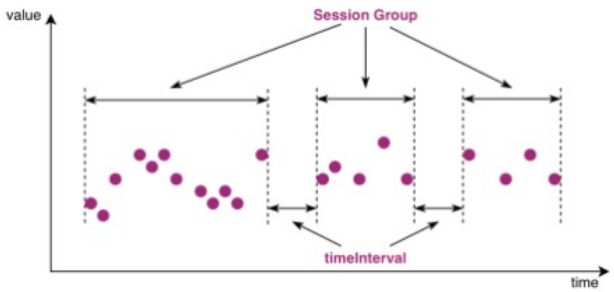
Downsampling Query



View the down-sampled original high-frequency data, remove unnecessary data details, and restore the basic trend of the data

(e.g. querying the average value of a wind turbine every 5 minutes in the past day)

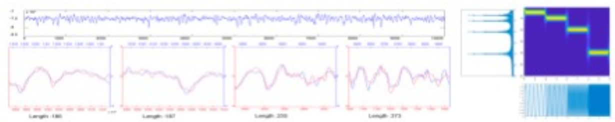
Sequence Segmentation Query



Perform multi-dimensional segmented queries based on the threshold of time series changes, interruption intervals, etc.

(e.g. querying the total operating time of a device over a certain period of time)

More Built-in Queries

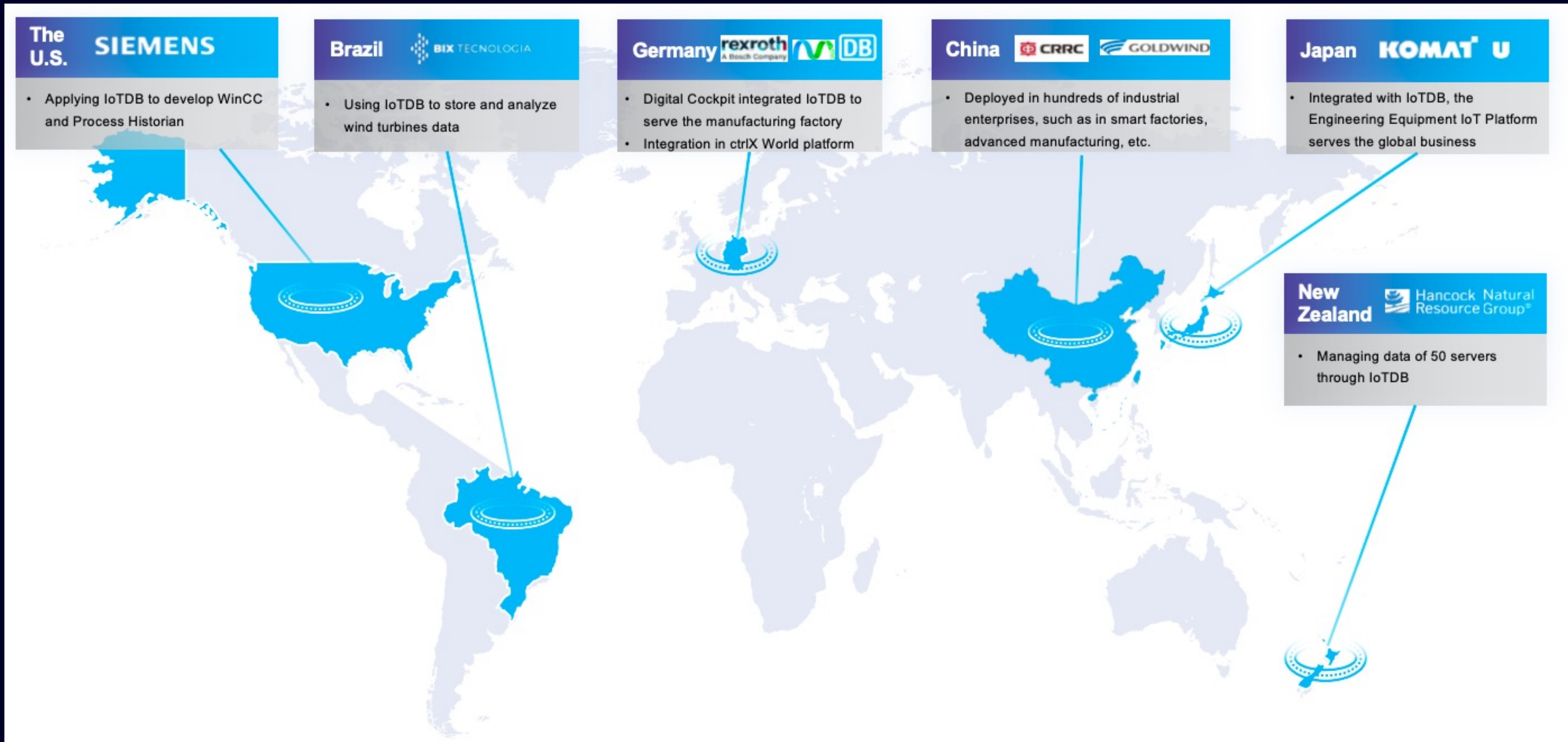


- **Built-in data processing functions:** quickly calculate the median, mean, repair missing data, aligning disturbed timestamps, etc.
- **UDFs：** data repair (ValueFill, TimestampRepair…), pattern matching(Cov, Dtw, Pearson…), etc.

SIEMENS

Use cases

Worldwide Use Cases



Scenarios Cover Space & Air & Ground & Sea

Deployment in Various Industries

BOSCH
rexroth A Bosch Company
SIEMENS
DB
pragmatic INDUSTRIES
MCC

Haier
Lenovo
NIO
PipeChina
GEELY
SINOPEC
AVIC

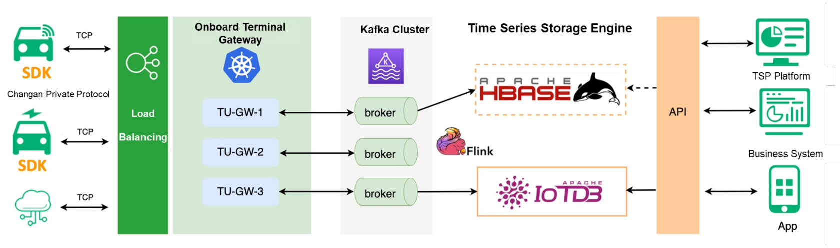
CRRC
CHANGAN
STATE GRID CORPORATION OF CHINA
COMAC
cnooc
CNPC
CGN

WEICHAI
Goldwind
WANHUA
CSSC
TAIJI 太极
CHINA TOWER 中国铁塔
BAOWU

KOMATSU
CNNC
Cloudwise
JD.COM
湖南大唐先一科技有限公司 Hunan Datang Xianyi Technology Co.,Ltd
CHINA HUANENG
• • •

Use Case in Automotive Industry

A query system for mass connected-cars status data



Highlights:

- Stores over 150 million time series
- 1 IoTDB server replaced 25 nodes of Hbase
- 12 millions data points write-in per second
- Data management of 500 signals from 200,000 connected vehicles
- Millisecond-level response-times for single vehicle time range queries and latest point queries

SIEMENS

Use Case in Metallurgy

An O&M platform for the full-scale data management in the entire group

[E4] Baowu Steel Group

- ❑ Plan: storing data at an hourly-level
  Fact: data stored at daily-level
- ❑ Plan: storing various types of data
  Fact: only relational data stored (no time series)

[E3] Site

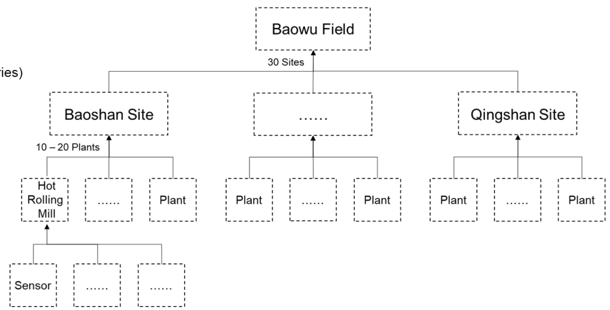
Plan: storing data at a second-level
Fact: data stored at a minute-level

[E2] Edge

Plan: storing data at the original frequency
Fact: data stored at a second-level

[E1] Embedded

Data collection and storage facilities are determined by OEMs



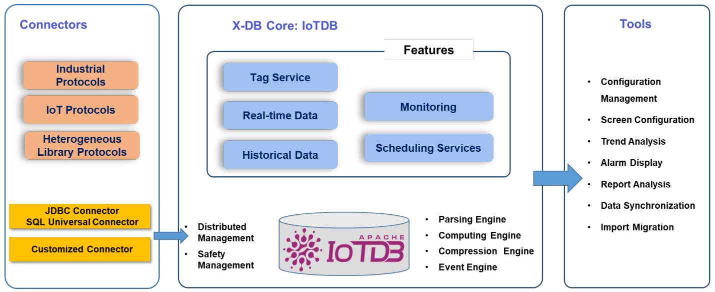
Highlights:

- ➢ Multi-level deployment, achieving an integrated 'edge-to-cloud' data management
- ➢ Data insertion rate over 30 million points per second
- ➢ Compression ratio is approximately 10:1
- ➢ Queries can cover up to 10 years of equipment data for downsampling analysis
- ➢ For a single time series with 200 billion data points, querying takes just a few seconds

SIEMENS

Use Case in Power Plants (Renewable Energy)

A new energy data management system was developed

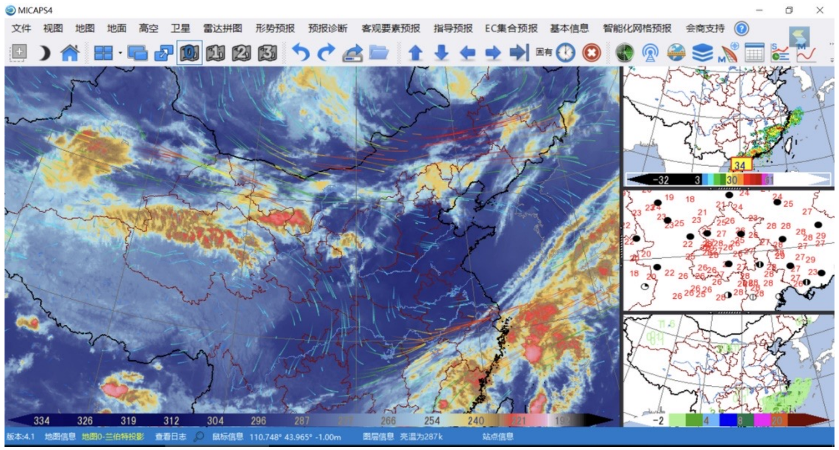


Highlights:

- The system serves over 60 power plants
- Over 1.7 billion points generated daily in each power plant
- Over 3 trillion points stored
- Reducing 95% O&M costs
- UDF and rich query functions enhanced the monitoring and analysis

Use Case in Public Sector

A new meteorological real-time data management system was developed



Highlights:

- The new generation of professional meteorological forecast system MICAPS4 was developed for daily weather forecasting affairs in China
- The data of more than 150,000 national ground observation stations was stored in Apache IoTDB
- The live data display and analysis capabilities were enabled
- The average accuracy of radar extrapolation forecasts was improved by 40%

Thank You for Listening!

SIEMENS

Contact

Published by Tsinghua University

Yuan Tian
Apache IoTDB PMC Member
Phone +86 195 1473 5670

E-mail jackietien97@gmail.com

SIEMENS

The Applications of IoTDB in Siemens Projects

Li Dong (T RDA FOA ART-CN1)

11/17/2023

SIEMENS

Outline

1

Historian for Experience Center

2

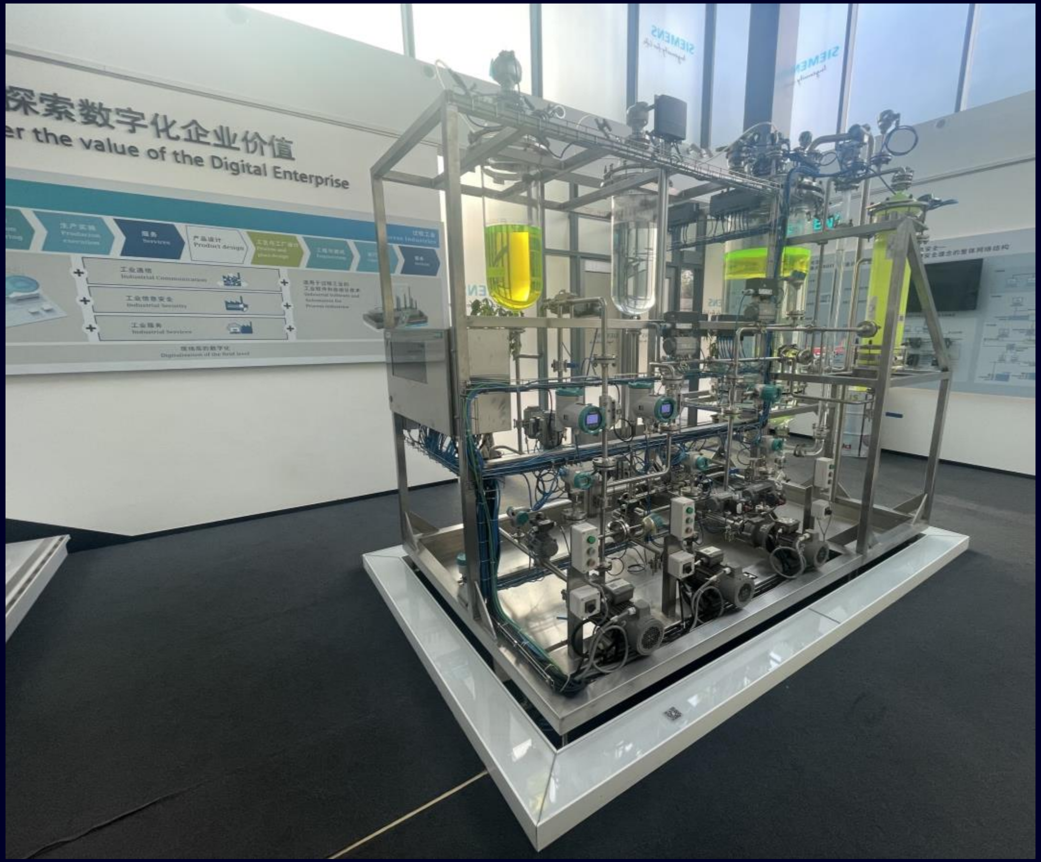
Enterprise Historian

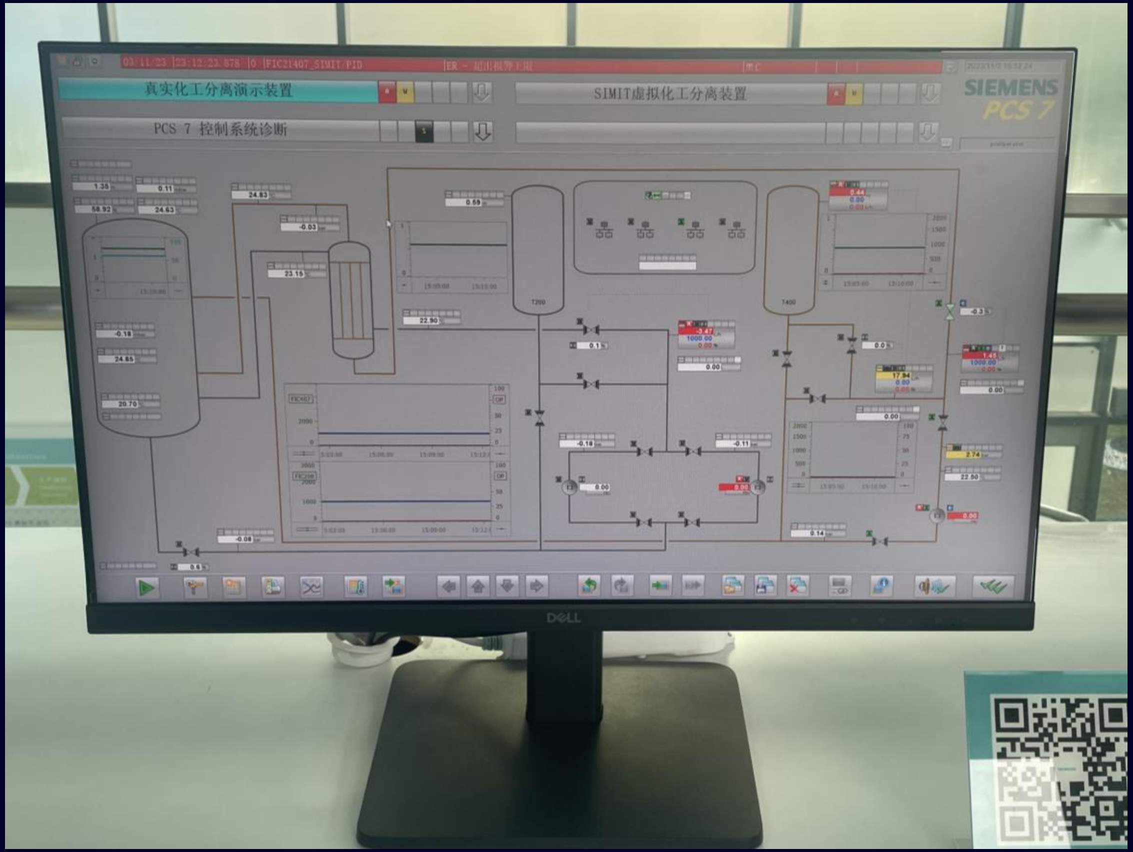
3

Future Historian

SIEMENS

Historian for Siemens Shanghai Yangpu Process Industry Digitalization Experience Center (RC-CN DI PA OEC DE-E DAO)





SIEMENS

Project Background

This project originated from the Control & Automation Campaign.

Current solution

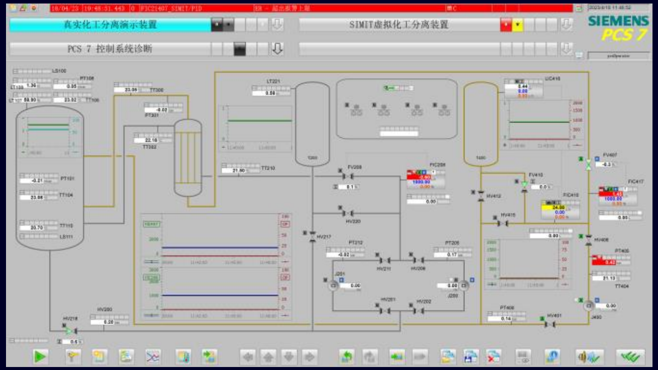
A time series database system is erected based OpenTSDB with HBase and Hadoop in Shanghai PA DEC (Digitalization Experience Center). The data from PCS 7 are transferred to this time series database by OPC UA client which is developed by an opensource Java program. Grafana is used to query and visualize the data in the database of OpenTSDB. So in this case, the basic functions of PIMS (data collection, store and visualization) are realized.

But the performance and reliability of OpenTSDB is not test, and it is not an easy job to maintain the big data system of Hadoop and HBase. The application of PIMS is just limited by Grafana.

Problems:
- Too heavy
- Difficult in installation
- Not stable, crashes sometimes

SIEMENS

IoTDB Solution



data

PCS 7 System with OPC UA Server

OPCUA IoTDB Gateway

Newly developed

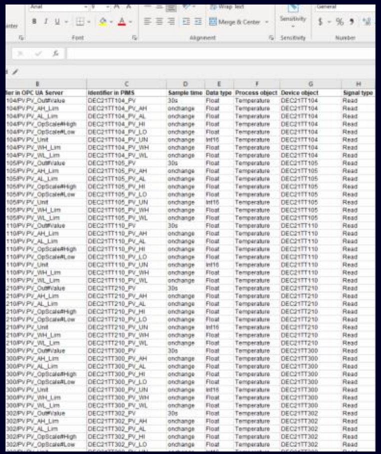
APACHE IoTDB

On-the-self

Grafana

Prometheus

configuration



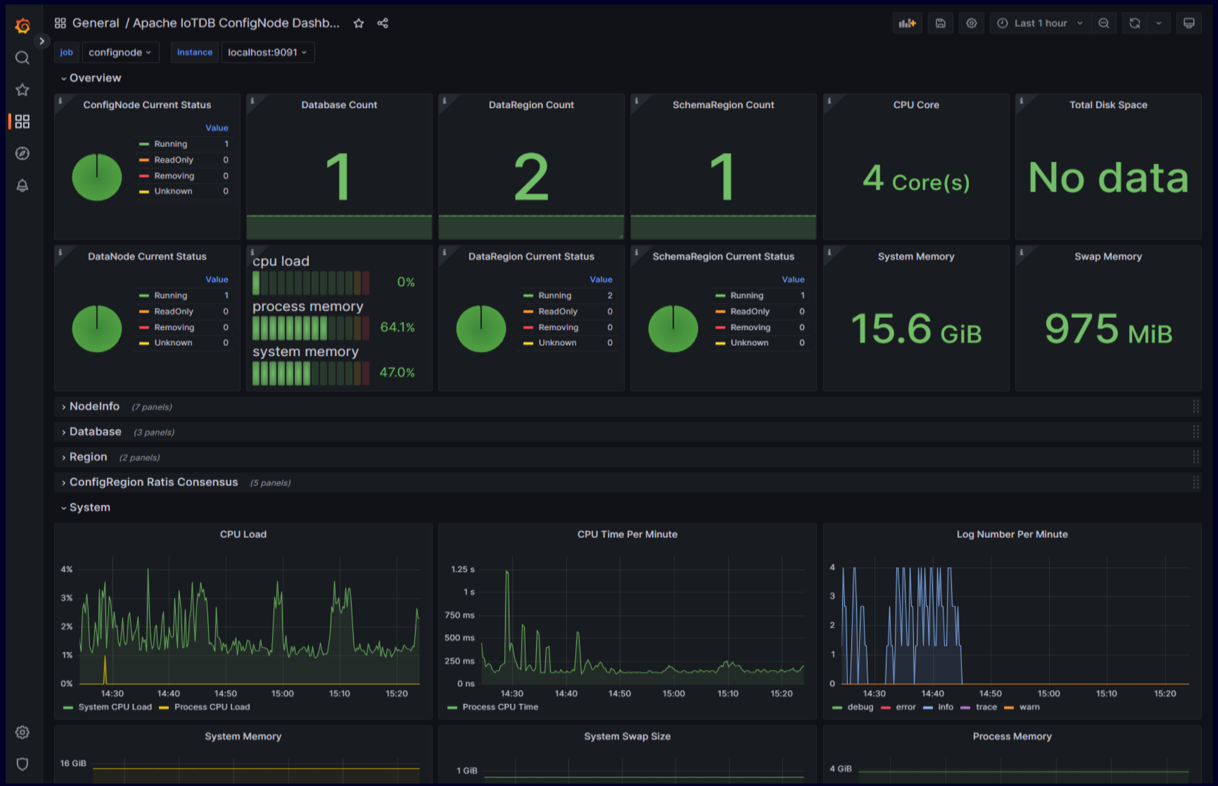
OPC UA data points（~300)

SIEMENS

Grafana dashboard

Business Monitor

System Monitor

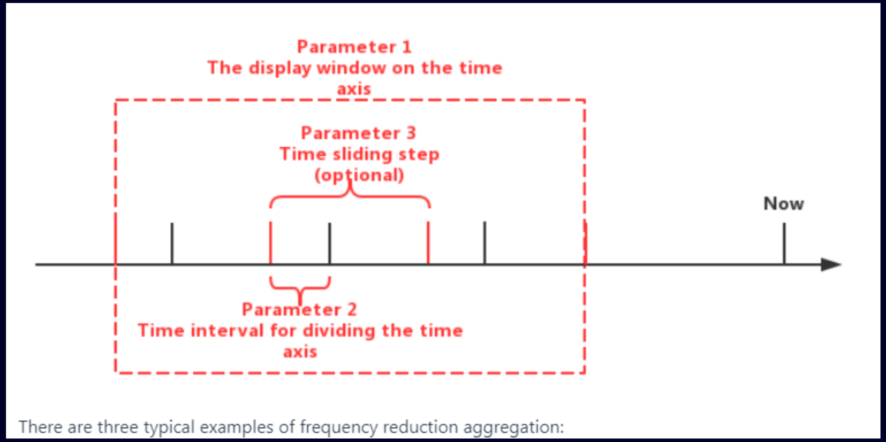


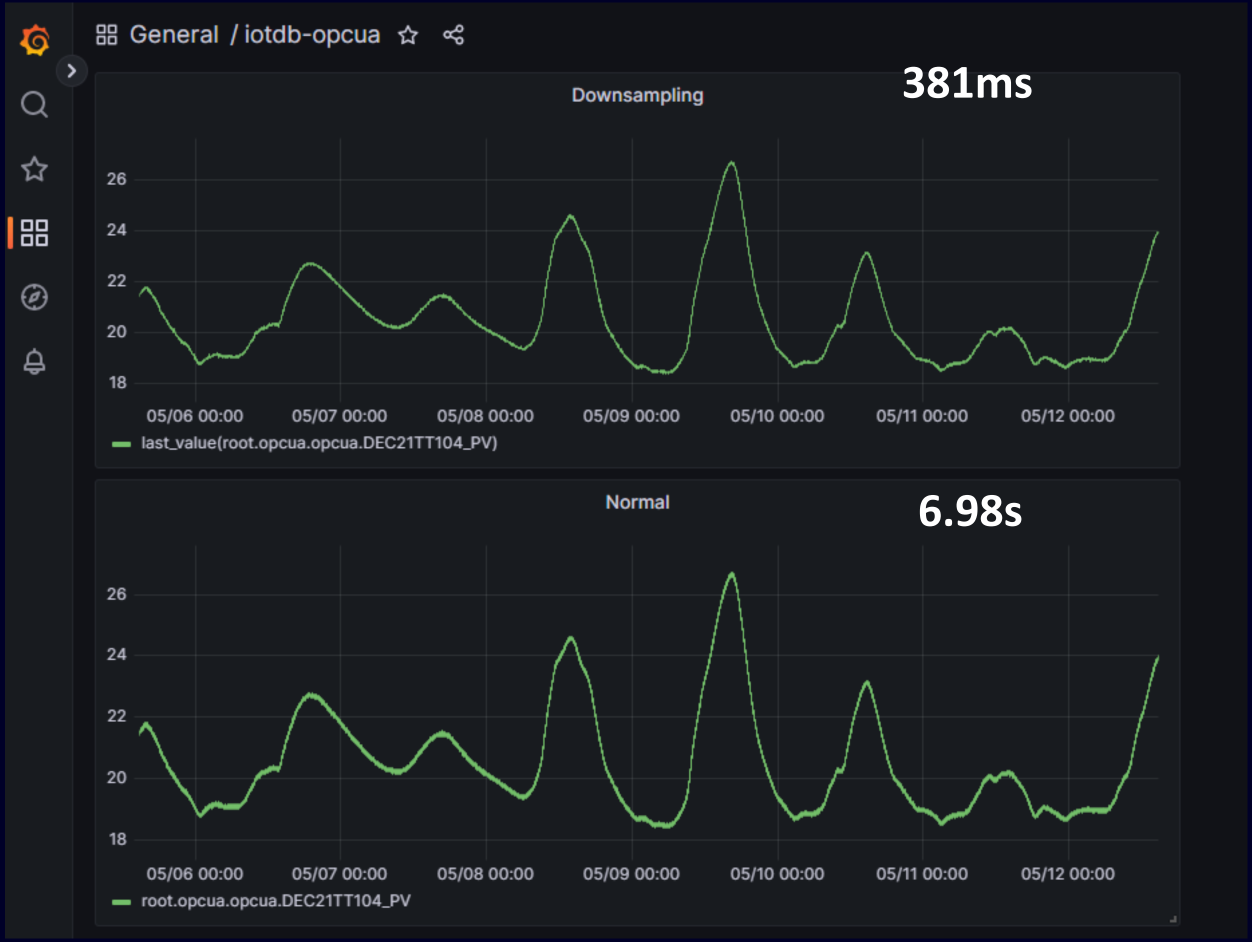
SIEMENS

Challenge: One query for 7-day data

A big data
- 15 panels
- 43 measurements
- 10,481,938 datapoints for 7-day
- estimated time consumption: 5-10mins

Optimization: Down sampling Aggregate Query
- Same trend in 7-day scale
- 18x faster query speed





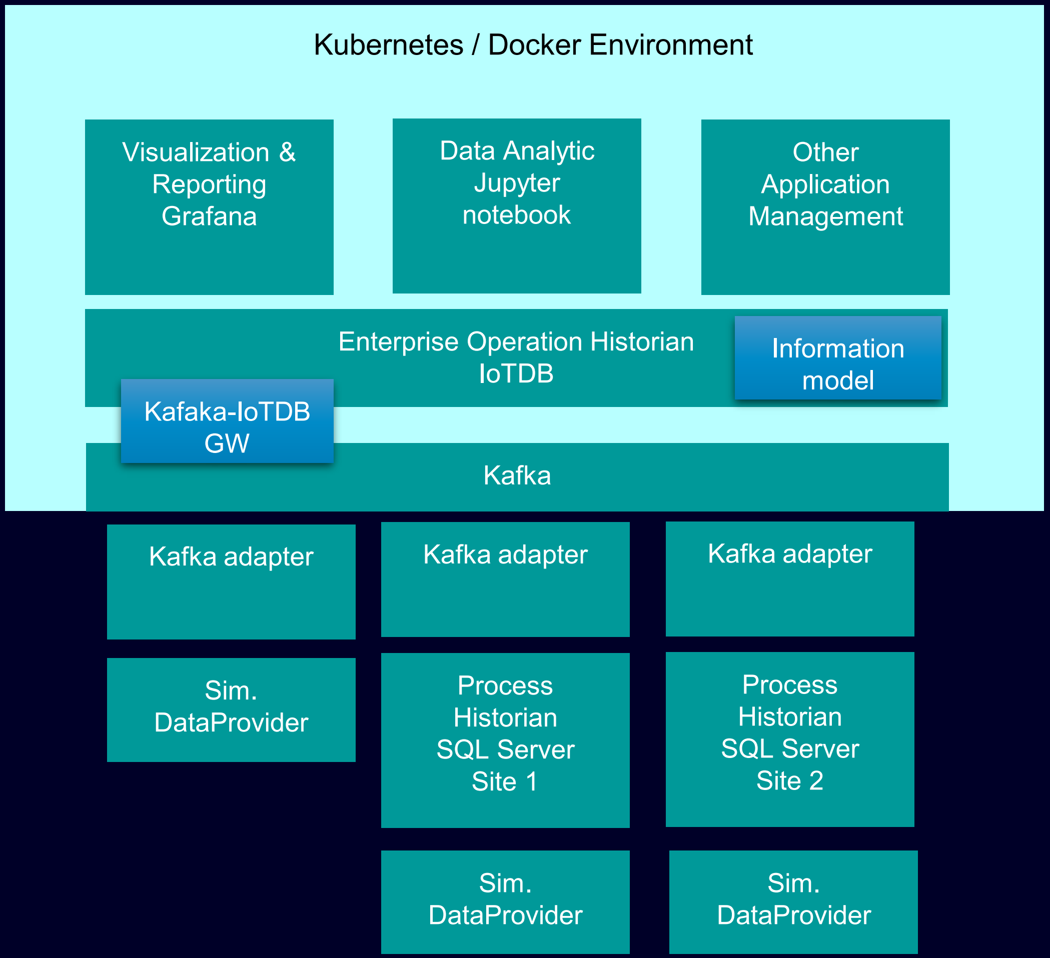
SIEMENS

Apache IoTDB is simple.

- Installation — On-the-self binary
- Connection — Pre-built Plugins
- Query — Aggregate functions

SIEMENS

Enterprise Historian (DI PA AE TI)

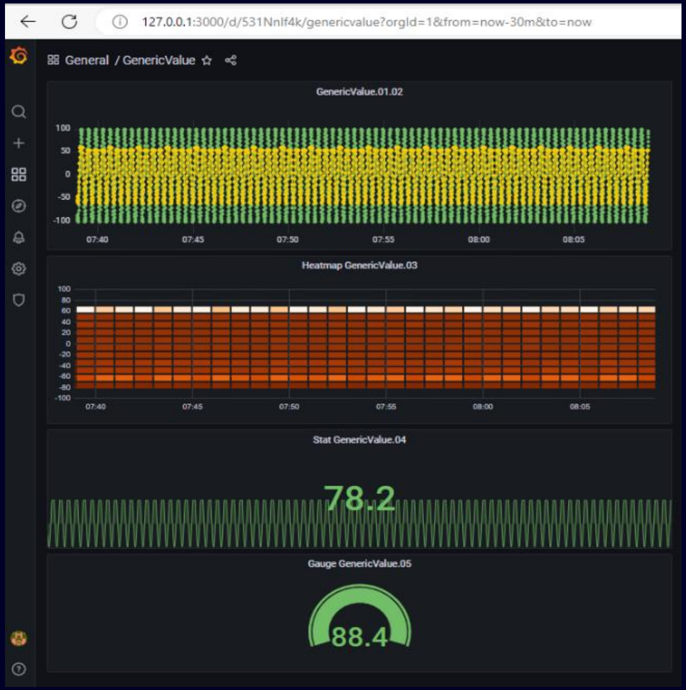


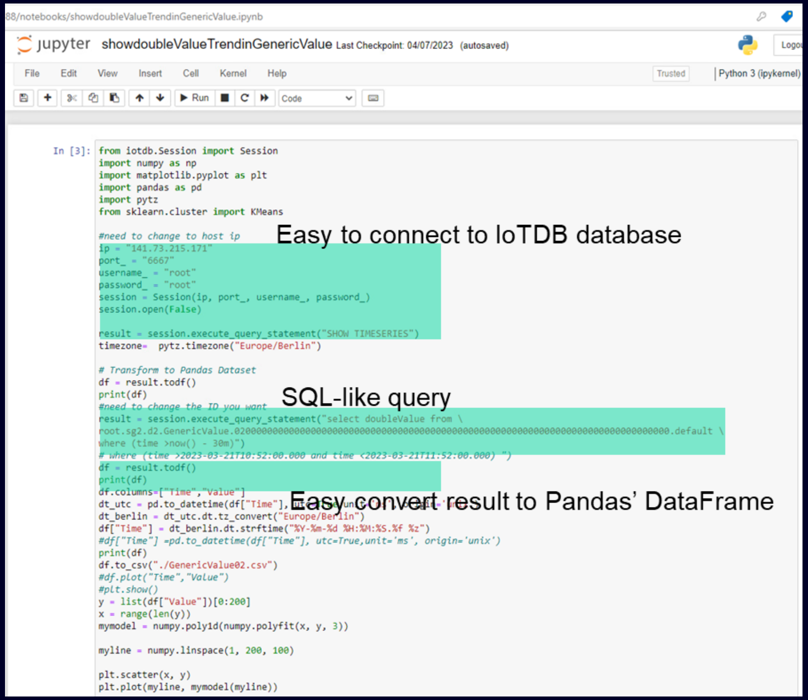
Overall Goal: Validate if IoTDB as a Enterprise wide Operational Historian for Process and IoT data

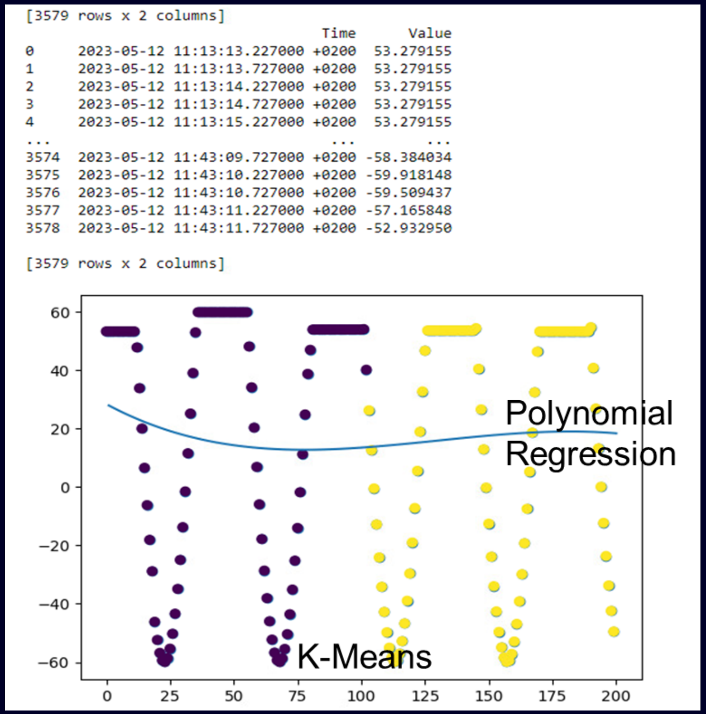
- Upload Data into IoTDB
  - (Tags, Events/messages)
  - Kafka based connection
  - Service to collect new Data every few seconds
  - from multiple PH Instances
  - from generic Data Provider
- Data Management
  - Build IoTDB Data Model
  - Show how to modify existing data tables with new column
  - Remove and reinsert time slices of data from/in IoTDB
- Data Access
  - Run Complex queries on IoTDB data / Compare with PH
  - Access data via Grafana
- Scalability / Deployment
  - With Kubernetes/Score cluster
  - Docker Based

SIEMENS

Open for Grafana and Python Jupiter







SIEMENS

Challenge: support multiple types for same data

Situation:

- A data can be of different types in database and can be changed at runtime.
- In relational databases, One column is an enumeration implying data type, another column is a string which need to convert to detail type.

In IoTDB:

- We did not use a string format to save data, which is not easy to data encoding and compression. For example, the string representation of floating-point data.
- Each type(text, int, double…) has a standalone timeseries.

timeseries	alias	storage group	dataType	encoding	compression	tags	attributes	deadband	deadband parameters
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.0.Type	null	root.sg2	INT32	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.0.textValue	null	root.sg2	TEXT	PLAIN	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.0.Quality	null	root.sg2	INT32	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.0.intValue	null	root.sg2	INT32	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.0.floatValue	null	root.sg2	FLOAT	GORILLA	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.0.booleanValue	null	root.sg2	BOOLEAN	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.0.Flags	null	root.sg2	INT64	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.0.doubleValue	null	root.sg2	DOUBLE	GORILLA	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.0.longValue	null	root.sg2	INT64	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.1.Type	null	root.sg2	INT32	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.1.textValue	null	root.sg2	TEXT	PLAIN	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.1.Quality	null	root.sg2	INT32	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.1.intValue	null	root.sg2	INT32	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.1.floatValue	null	root.sg2	FLOAT	GORILLA	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.1.booleanValue	null	root.sg2	BOOLEAN	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.1.Flags	null	root.sg2	INT64	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.1.doubleValue	null	root.sg2	DOUBLE	GORILLA	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.1.longValue	null	root.sg2	INT64	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.default.Type	null	root.sg2	INT32	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.default.textValue	null	root.sg2	TEXT	PLAIN	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.default.Quality	null	root.sg2	INT32	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.default.intValue	null	root.sg2	INT32	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.default.floatValue	null	root.sg2	FLOAT	GORILLA	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.default.booleanValue	null	root.sg2	BOOLEAN	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.default.Flags	null	root.sg2	INT64	RLE	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.default.doubleValue	null	root.sg2	DOUBLE	GORILLA	SNAPPY	null	null	null	null
root.sg2.d2.runtime.01000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.default.longValue	null	root.sg2	INT64	RLE	SNAPPY	null	null	null	null

SIEMENS

Challenge : Detect duplication and Save all the versions

Situation:

- There are disorderly (late than current time) duplicate (same timestamp and field ID) data in the system that needs to be updated, and old data also needs to be retained.
- In relational databases, this will trigger primary key exceptions (PH) and re-store data in foreign key association tables.

In IoTDB:

- Ensure the most common case is the most efficient (storage and throughput)
  - Sequence data (Latest timestamp) 99%
  - Disorder data
  - Disorder data with duplicated timestamp
- Each data version(default, 0, 1…) has a standalone timeseries, only adding multiple modification records for specific timestamps (No need for sparse matrices)
- Always get latest data in default query

```
IoTDB> select * from root.sg2.d2.runtime.010000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.** where time = 2023-02-10T18:22:33.214 ALIGN BY DEVICE
```

Time	Device	Type	textValue	Quality	intValue	floatValue	booleanValue	Flags	doubleValue	longValue
2023-02-10T18:22:33.214+08:00	root.sg2.d2.runtime.010000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.0	2	null	106	53	null	null	0	null	null
2023-02-10T18:22:33.214+08:00	root.sg2.d2.runtime.010000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.1	2	null	106	59	null	null	0	null	null
2023-02-10T18:22:33.214+08:00	root.sg2.d2.runtime.010000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.2	0	hello,world	9	null	null	null	2	null	null
2023-02-10T18:22:33.214+08:00	root.sg2.d2.runtime.010000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.3	11	null	71	null	null	null	6	70.34891043007902	null
2023-02-10T18:22:33.214+08:00	root.sg2.d2.runtime.010000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.4	0	hello,world	90	null	null	null	6	null	null
2023-02-10T18:22:33.214+08:00	root.sg2.d2.runtime.010000000000000000000000000000000000000000000000000000000000000000000000000000ThreadID0AA9.default	0	hello,world	90	59	null	null	6	70.34891043007902	null

SIEMENS

"

Apache IoTDB is open. Easy to adapt to our business logic:

• Multiple types	each type as a timeseries
• Duplicated Data	each version as a timeseries

SIEMENS

Future Historian (DI FA HMI)

Customer, Cooperator, Timeline

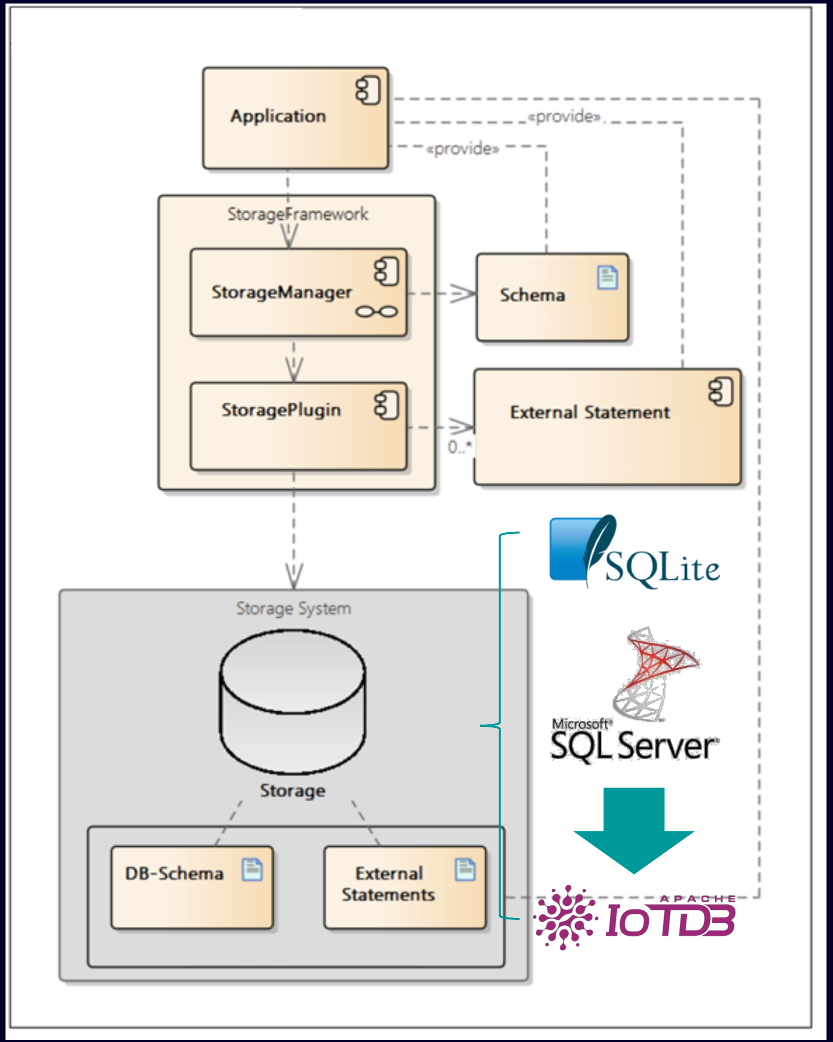
- DI FA HMI, ART-US and ASY-CN.
- Part of a long term project from FY20/21 to now.

Project Scope for FY23

- Develop POC of IoTDB Plugin for IOWA Logging Service.
- Evaluate IoTDB as high-performance database and possible MSSQL replacement candidate.

WinCC Advanced / Comfort

WinCC UA



SIEMENS

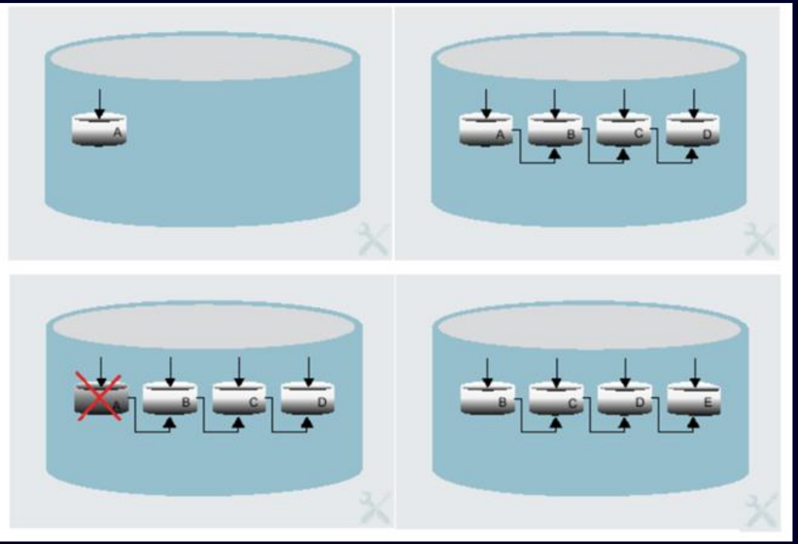
Challenge : Back up and Save storage media

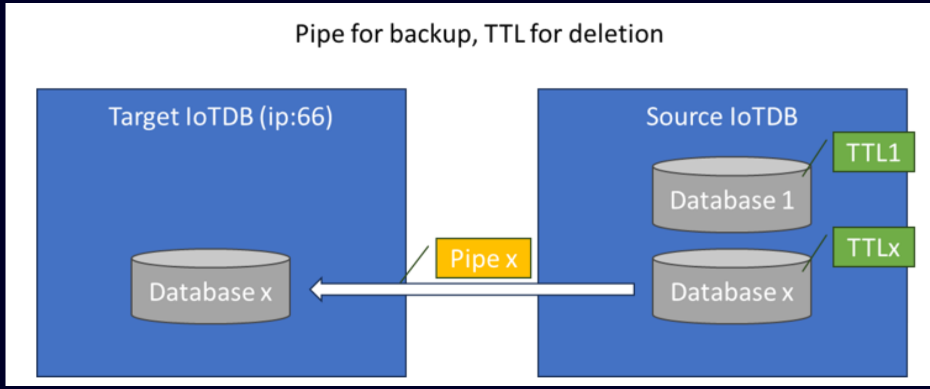
Situation:

- The business needs to regularly backup data and delete old data.
- Different data groups have different storage periods in local.
- In current solution, different data is saved in different folders based on data groups and time periods. The folders can be manually backup and deleted.

In IoTDB:

- We could use Pipe and Time To Live (TTL) to control the total disk space usage and prevent the machine from running out of disk space.





SIEMENS

Tiered storage

- Enterprise IoTDB also support Tiered storage which allows users to define multiple layers of storage, spanning across multiple types of storage media (Memory mapped directory, SSD, rotational hard discs or cloud storage). Users can classify data based on its hot or cold nature and store data of different categories in specified "tier".

tier	data path	data range	threshold for minimum remaining disk space
tier1	path 1：/data1/data	data for last 1 day	20%
tier2	path 1：/data2/data path 2：/data3/data	data from past 1 day to past 10 days	15%
tier3	Remote AWS S3 Storage	data from 1 day ago	10%

SIEMENS

Dual active deployment

- Two independent IoTDBs with completely independent configurations that can simultaneously receive external writes. Each independent cluster can synchronize the data written to itself to another cluster. It can be used for disaster recovery.
- Two instances can form a high availability group: when one instance stops serving, the other instance is not affected. When the stopped instance starts again, another instance will synchronize the newly written data.

```
create pipe atob
with extractor (
    'extractor'='iotdb-extractor',
    'extractor.forwarding-pipe-requests'='false'
)
with connector (
    'connector'='iotdb-thrift-connector',
    'connector.node-urls'='<B集群地址>'
);

start pipe atob;
```

```
create pipe btoa
with extractor (
    'extractor'='iotdb-extractor',
    'extractor.forwarding-pipe-requests'='false'
)
with connector (
    'connector'='iotdb-thrift-connector',
    'connector.node-urls'='<A集群地址>'
);

start pipe btoa;
```


SIEMENS

Apache IoTDB is high reliable.

- Backup
- Redundancy

Pipe/TTL/ Tiered Storage

Dual active deployment

SIEMENS

Benefits for IT and OT integration:

Simple,
Open
and Reliable.

SIEMENS

Contact

Published by Siemens T

Dong Li
Senior Key Expert
T FOA ART-CN1
Siemens Ltd., China
No.7, Wangjing Zhonghuan Nanlu.
Chaoyang District, Beijing 100102, P.R.China
E-mail lidong-ld@siemens.com

SIEMENS