KubeCon CloudNativeCon

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

AI_dev

Open Source GenAI & ML Summit

China 2024

KubeCon

CloudNativeCon

THE LINUX FOUNDATION

China 2024

Time Series Database on Kubernetes: Efficient management of massive Internet of Vehicles data

About Me

Vicky Lee

I’m a Time-series database expert in the HUAWEI CLOUD Database Innovation Lab and the Co-founder of the openGemini community, has been engaged in distributed databases and NoSQL databases as a cloud service for many years. Currently, mainly dedicated to openGemini development

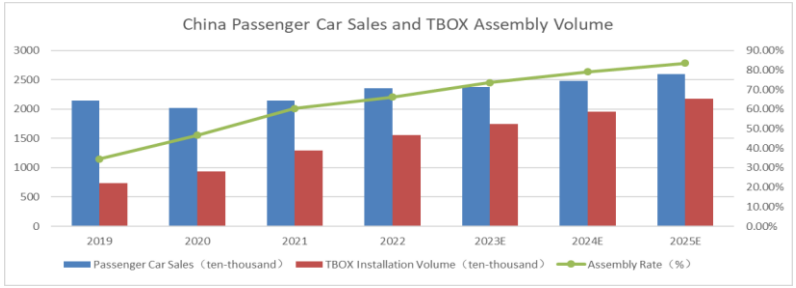
Content

- IoV Technology Architecture Evolution and Industry Challenges
- IoV Solution Based on openGemini
- openGemini Key Technologies

IoV Technology Architecture Evolution and Industry Challenges

Technical Challenges: Performance, Real-time Analysis And Costs

More and more vehicles connected to internet



More than **70%** new vehicles have the network connection capability, and the annual IoV access increment exceeds **30%.**

Data collection frequency is becoming increasingly higher

GB Standard	Data: 100 + Columns Frequency: 30s
Enterprise Standard	Data: **1000 + Columns** Frequency: **10S/1S/100ms**

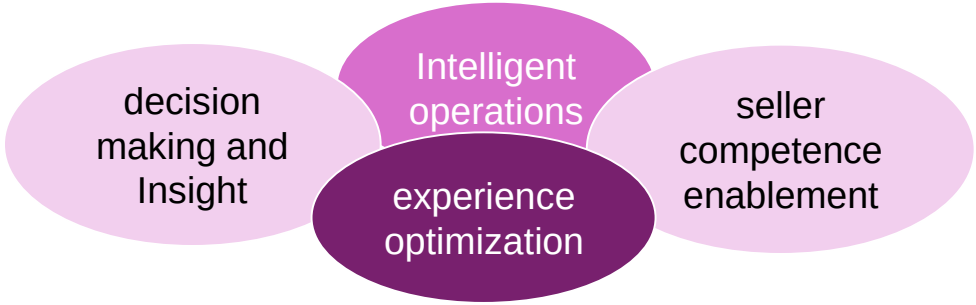
The enterprise standard data is more than **10x** that of the GB standard, it means that these data need to be quickly written into the database. If millions of vehicles report data， the write traffic reaches **10 GB/s**

Increasing storage costs

6 months	**3 years / 10 years / Permanent**
1 month	1 year
real-time analysis	offline analysis

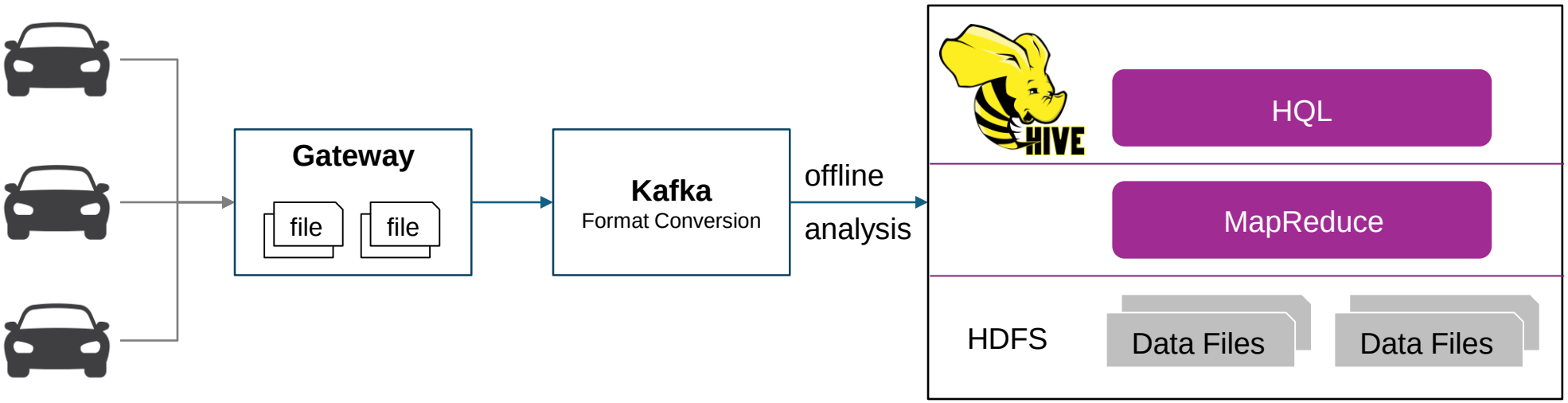
Massive data and long storage time. 1 million vehicles generate 3 PB data in a year.

Increasing requirements for real-time analytics



These scenarios, such as alarm, vehicle status query, and fault analysis, have higher requirements on real-time services

Early IoV Technical Architecture



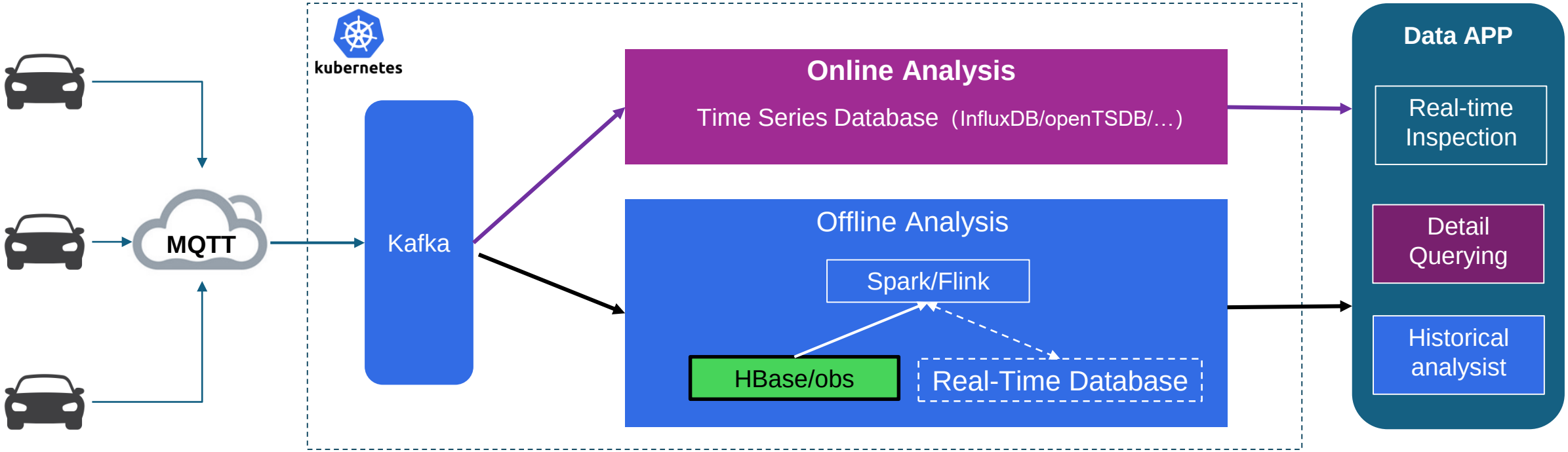
Benefits

1. Big data storage
2. Statistical Analyses

Challenges

1. File batch processing and low efficiency
2. Not meeting the requirements of real-time data analysis

Modern IoV Technical Architecture



Benefits

1. **K8s** enhances the resource utilization of computational frameworks and analysis tasks.
2. Big data technologies have greatly improved data processing efficiency.
3. **Time Series Databases** enable real-time analysis and querying capabilities for the IoV.

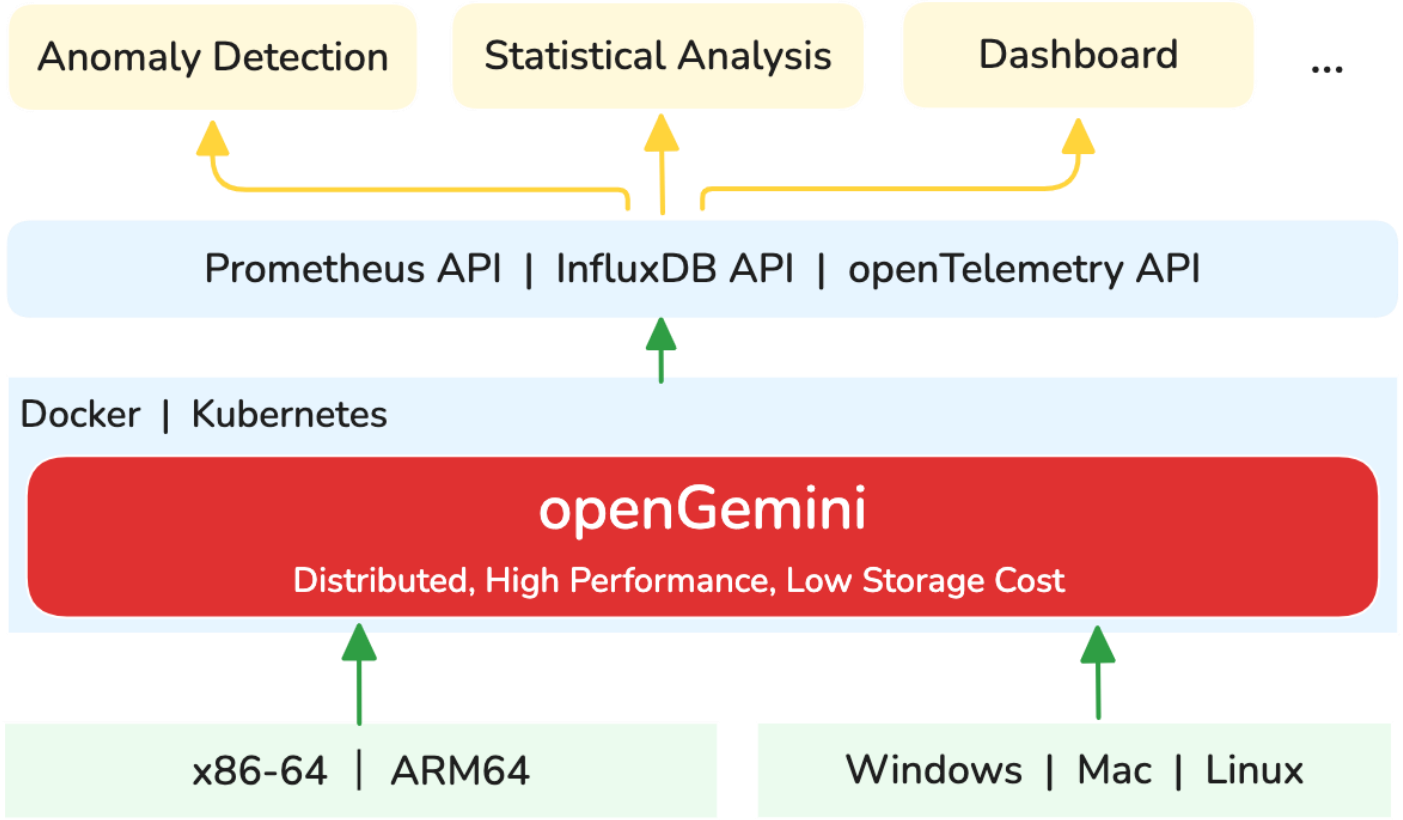
Challenges

1. The traditional time series database cannot meet the performance requirements of massive data writing and real-time data analysis.
2. HBase lacks effective data compression for long-term data retention, resulting in high data storage costs

IoV Solution Based on openGemini

What is openGemini

openGemini is a CNCF Time Series Database project, Focusing on the storage and analysis of massive observability data



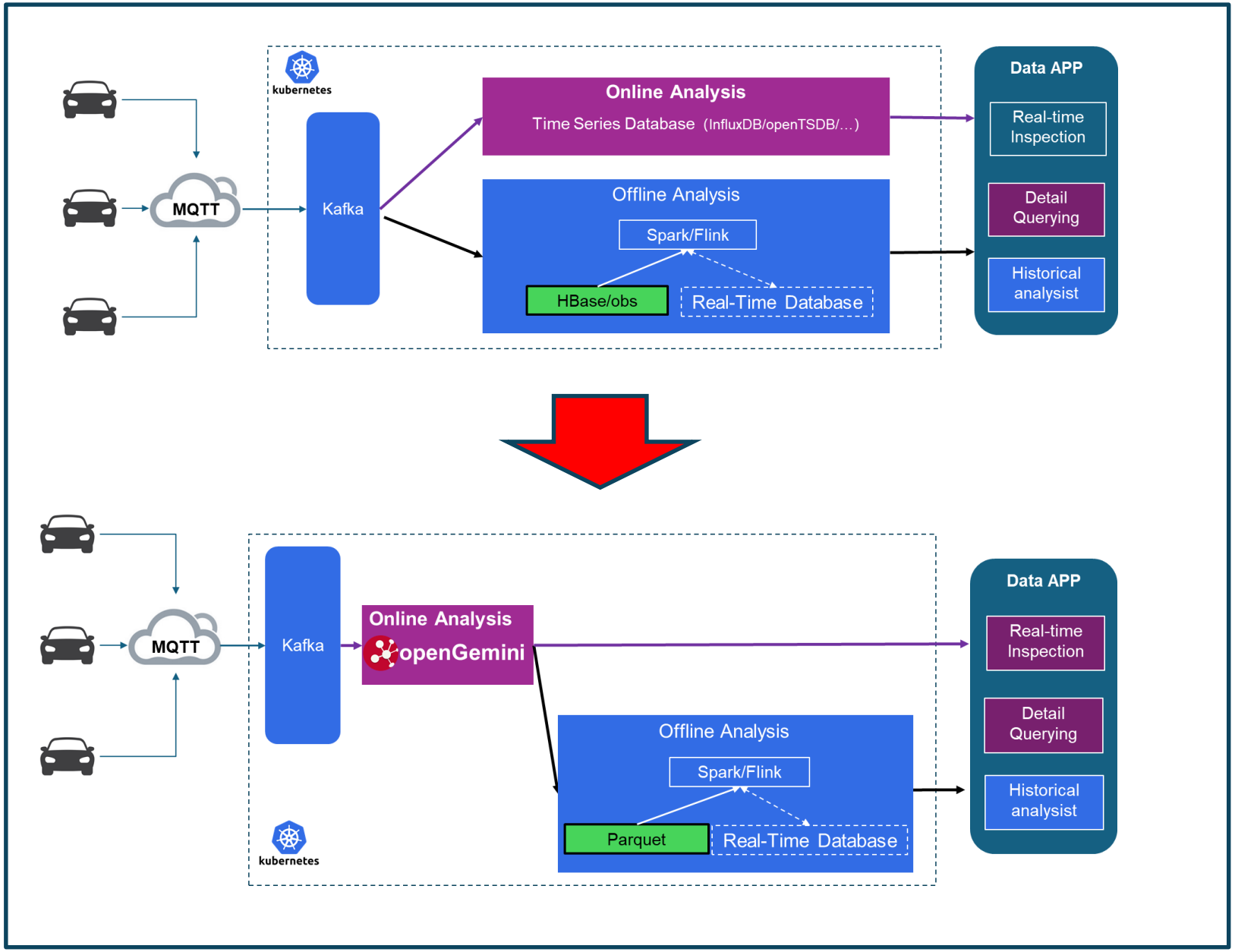
Features

- **Distributed**
- **High Performance**
- **Low Storage Cost**

applicable scenarios

- **IoT**
- **IoV**
- **Devops**
- **…**

openGemini： New Technical Architecture For IoV Massive Data



Benefits

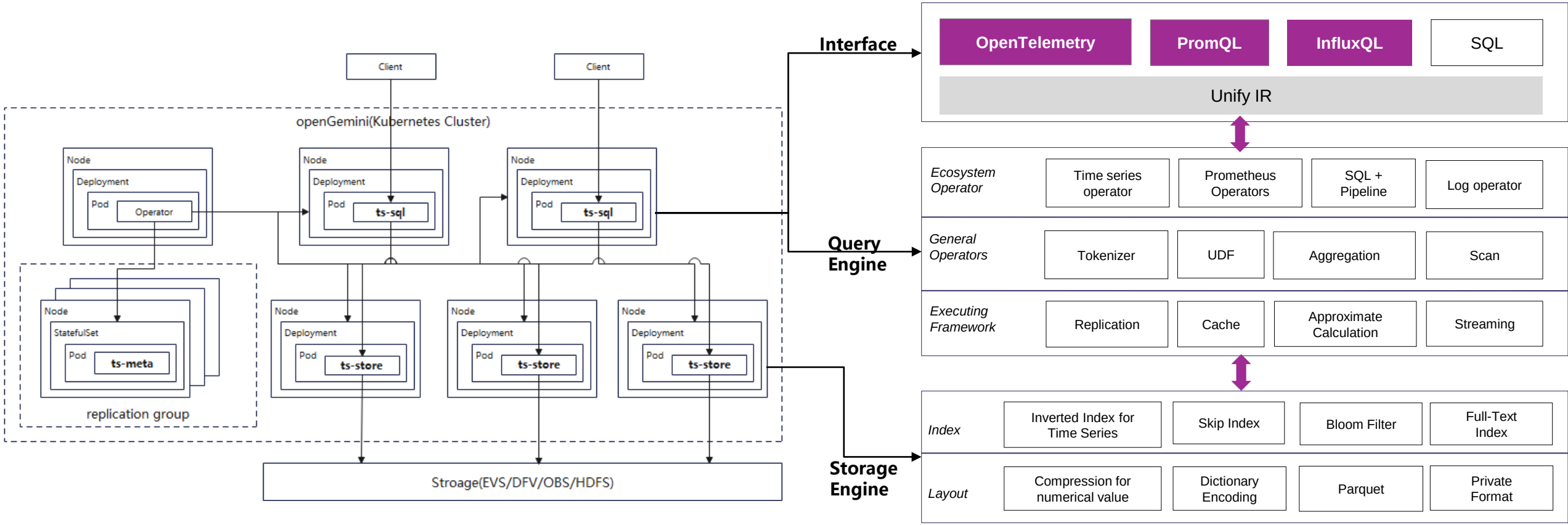
Fast Write
support millions of vehicles writing 10GB/s

Fast Query
data query response **Millisecond-level**

Low Cost Storage
reduce data storage costs **30%+**

openGemini Key Technologies

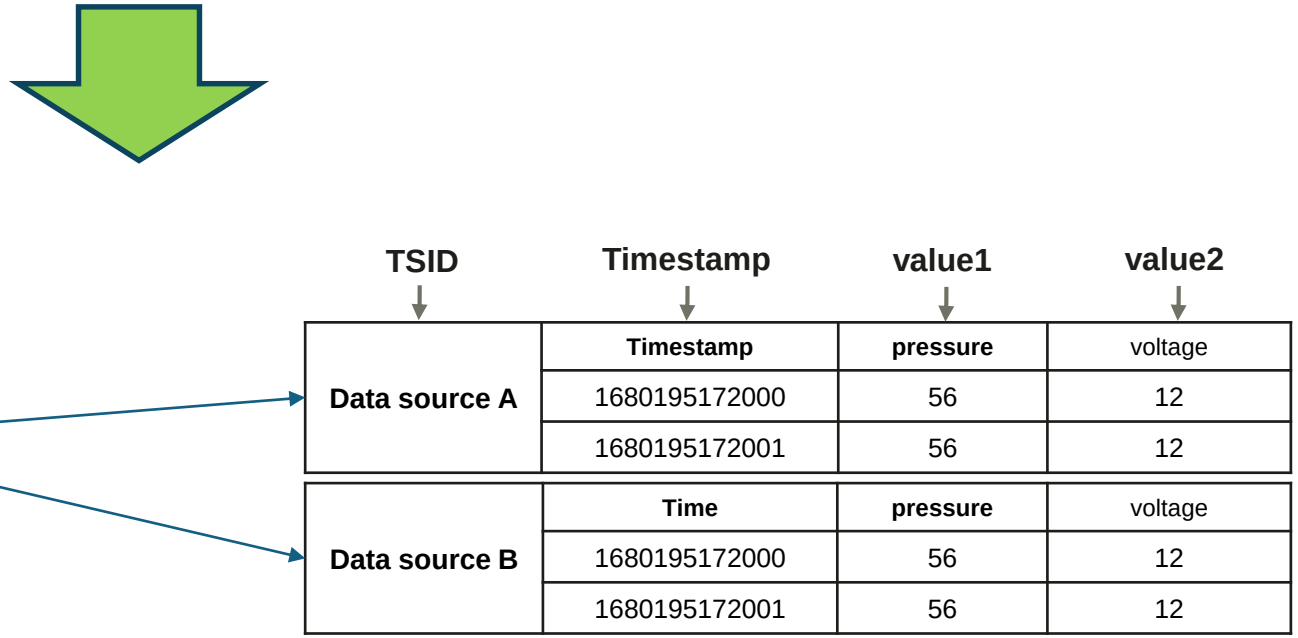
openGemini + Kubernetes



- **MPP (Massively Parallel Processing) Architecture**, Improve data processing performance by spreading workloads across multiple nodes
- **Scale Horizontally**, Increase the number of nodes to resist the impact of concurrent traffic.
- **Flexibility**, ts-sql and ts-store can scale-out independently

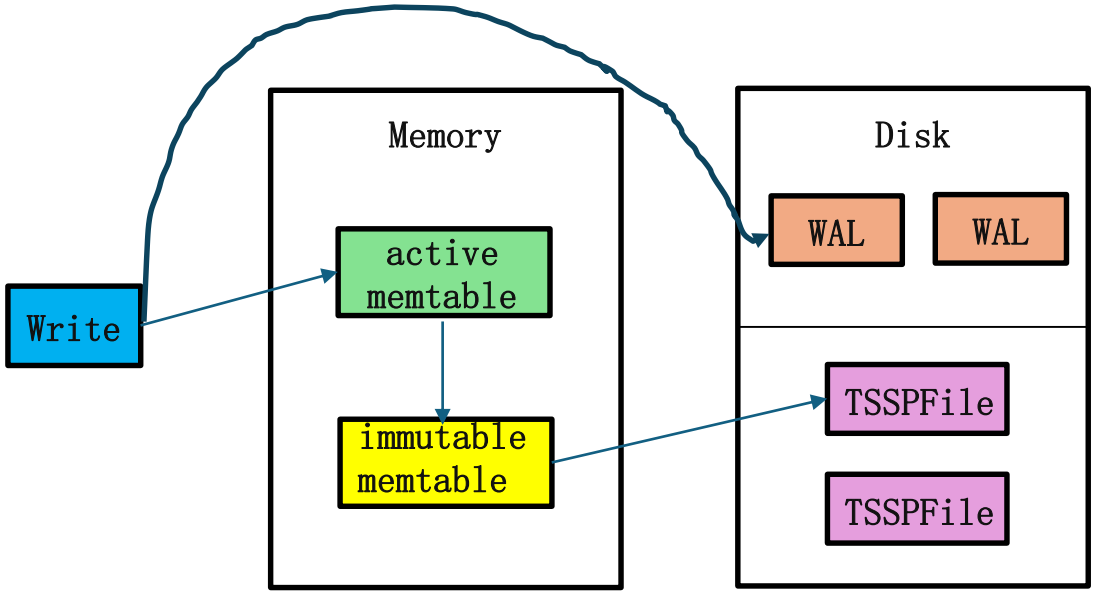
What is Time Series

Table	Tag	Field	Timestamp
vehicle	vin="1G1BL52P7TR115520" type="x3"	pressure=56 voltage=12	1680195172000
vehicle	vin="1G1BL52P7TR115521" type="x3"	pressure=58 voltage=11	1680195172000
vehicle	vin="1G1BL52P7TR115520" type="x3"	pressure=56 voltage=12	1680195172001
vehicle	vin="1G1BL52P7TR115521" type="x3"	pressure=57 voltage=13	1680195172001

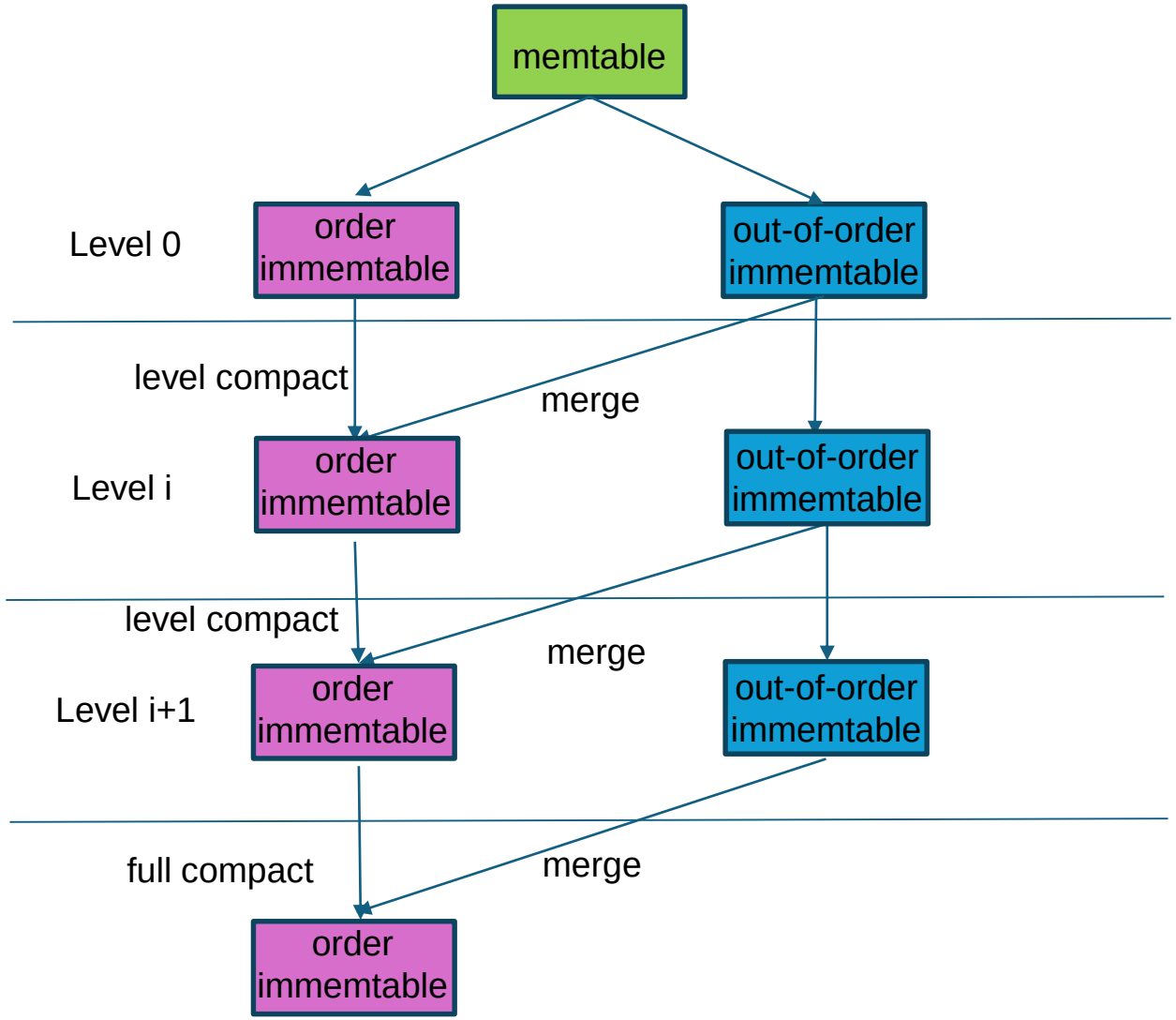


Data Layout

Why Write Data So Fast

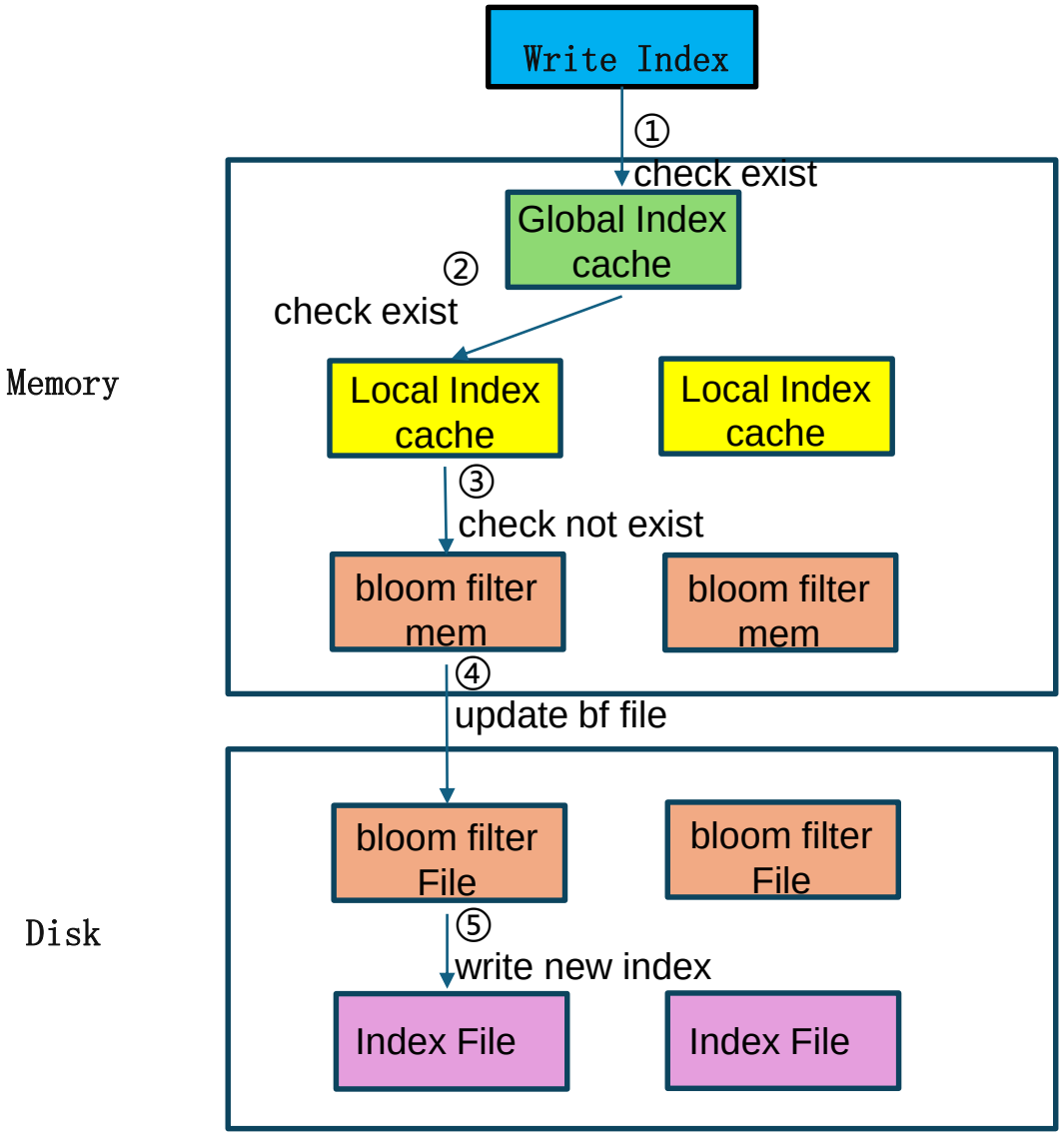


Write memTable and return



Delay merge

Why Write Index So Fast



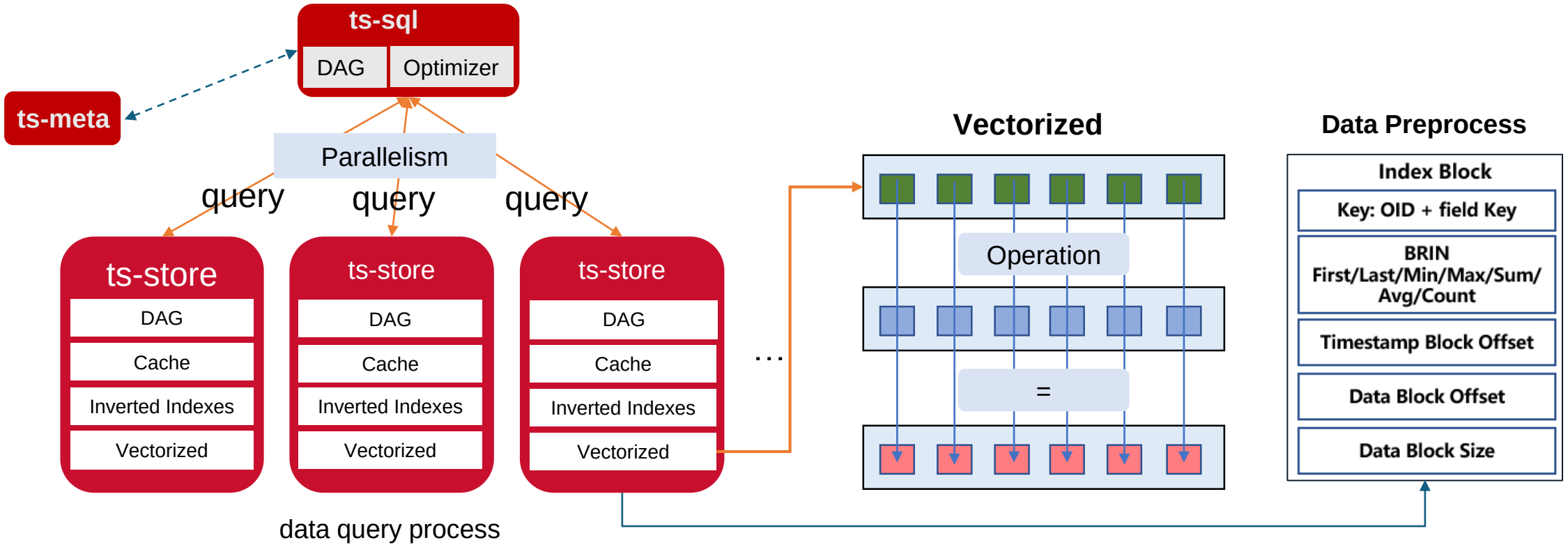
Benefits

1. Global-Local cache architecture improve cache hit ratio and reduce memory use
2. Bloom filter reduce the cost of index check.

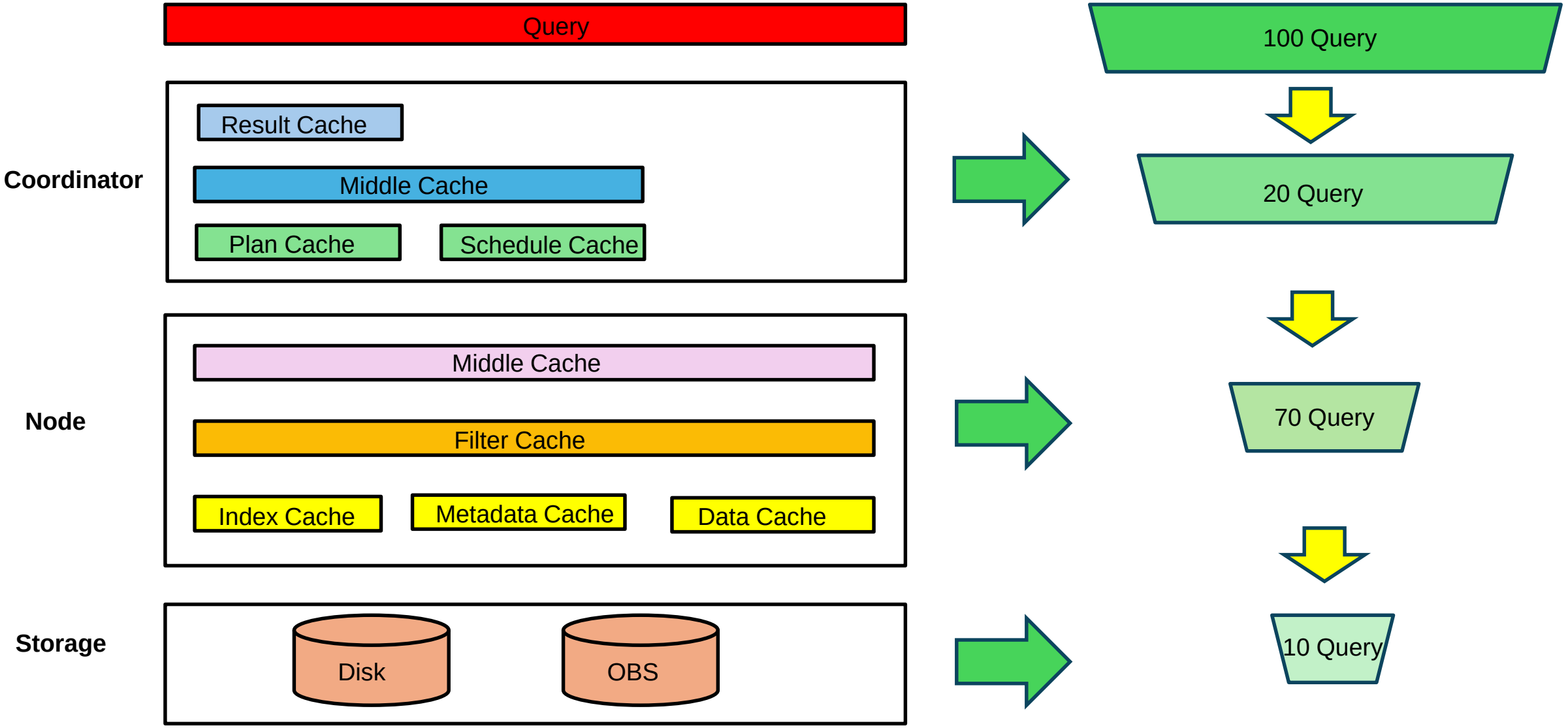
Query Data： Vectorized, Parallel Computing, Data Preprocess

```
SELECT vin,pressure,voltage from vehicle where vin='0' AND
time > now() – 1h and (pressure<50 or voltage<10) group by *
```

Make use of the parallel computing advantages of the architecture, faster response



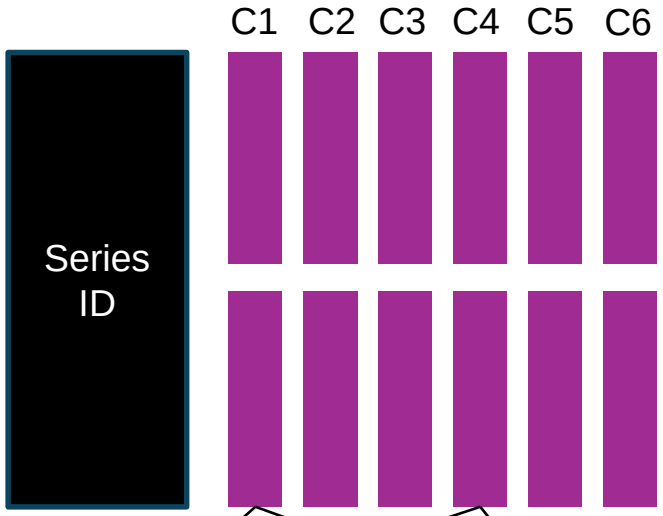
Why Query So Fast



Data Compression: 10x higher data compression efficiency

openGemini vs openTSDB（HBase）

column-based storage，Data from the same timeline is stored together



Data Type + Data Distribution

Data Compression Algorithms

Simple8b, Delta, Delta-of-Delta，RLE, Zigzag, Zstd, Snappy, Bit-packing, L4 …

Key	CloumnFamily	Qualifier	Timestamp	Type	Value
rowkey1	Personal_info	Name	T1	put	Peter
rowkey1	Personal_info	City	T2	put	Chicago
rowkey1	Personal_info	Phone	T3	put	132xxxxx
… Data Redundancy		…	…	…	…
rowkey2	Company_info	Name	T4	put	Qtime
rowkey2	Company_info	City	T5	put	Hong Kong

Data Compression Algorithms

Only Four: GZ，SNAPPY，LZO，LZ4

IoV data: Compared with HBase, openGemini has **10x** higher data compression efficiency.

Metric Store Evaluation

Compared with InfluxDB, openGemini has better read/write performance and data compression

Data Model	Specifications	Query Model Name	Time-series	concurrency	average delay by query (ms)		
					openGemini	InfluxDB	openGemini/InfluxDB
Devops	Single-node 32U128GB	single-groupby-1-1-12	300,000	32	5.61	22.93	409%
		single-groupby-1-1-1		32	2.02	4.20	208%
		single-groupby-1-8-1		32	4.10	11.72	286%
		single-groupby-5-1-12		32	9.72	95.04	978%
		single-groupby-5-1-1		32	2.79	11.66	418%
		single-groupby-5-8-1		32	5.91	46.60	788%
		cpu-max-all-1		32	3.74	13.55	362%
		cpu-max-all-8		32	9.34	88.19	944%
		double-groupby-1		8	21558.69	243356.06	1129%
		double-groupby-5		8	51607.11	OOM	–
		double-groupby-all		8	88777.61	OOM	–
		lastpoint		8	5501.98	OOM	–
		groupby-orderby-limit		2	9014.86	OOM	–
		high-cpu-1		32	11.30	29.08	257%
		high-cpu-all		1	32622.80	OOM	–

Data Model	Specifications	Database	Time-series	concurrency	Write performance	Disk Usage	Raw Data
					rows/sec		
Devops	Single-node 32U128GB	openGemini	300,000	32	**469,881**	11GB	259 million rows of data, 842 GB (text file)
		InfluxDB			**73,529**	14GB	

Reference: https://docs.opengemini.org/zh/guide/introduction/performance.html

Welcome To Try And Give Feedback

```
docker run –d –p 8086:8086 –name openGemini-dev openGemini-server
```

https://github.com/openGemini

https://www.openGemini.org

微信搜一搜 openGemini

https://join.slack.com/t/opengemini/shared_invite/zt-2naig1675-x3bcwgXR_Rw5OwDU5X~dUQ

slack