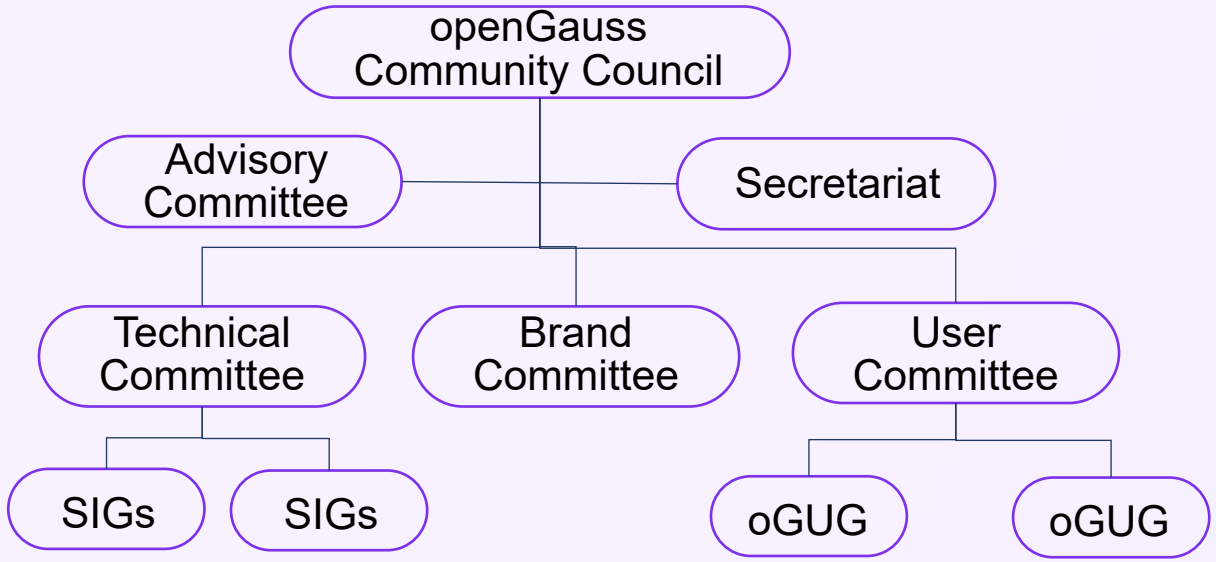
openGauss:
An Open-Source Database for Data Infrastructure

openGauss: an Open-Source Database for Data Infrastructure

openGauss is a multi-model database management system. It was officially open-sourced on June 30, 2020 and released under the Mulan Permissive Software License v2.

openGauss Community Organizational Structure



22
Council Members

28
SIGs

900+
Member Organizations

8,600+
Contributors

openGauss "1+2" Strategy

- Building high-quality databases, consolidating core capabilities, and cementing the cornerstone of partner trust
- Building oGRAC multi-read/write and SuperPoD databases, leading technological directions through architectural innovation
- Creating an AI-native multi-modal database foundation to unlock the intelligent potential of data elements

https://opengauss.org/en/member/

openGauss official account

openGauss

openGauss Community: Fostering Consensus via Openness, Forging a Mature Ecosystem through Collaboration

Dynamic Community Evolution

Gathering forces for collaborative open source

6 million+ Downloads	**900+** Member Organizations	**8,600+** Contributors
28 SIG	**30,900** Pull requests	**623,500** Review & Comments

Powering smart infrastructure upgrades

Better search capability

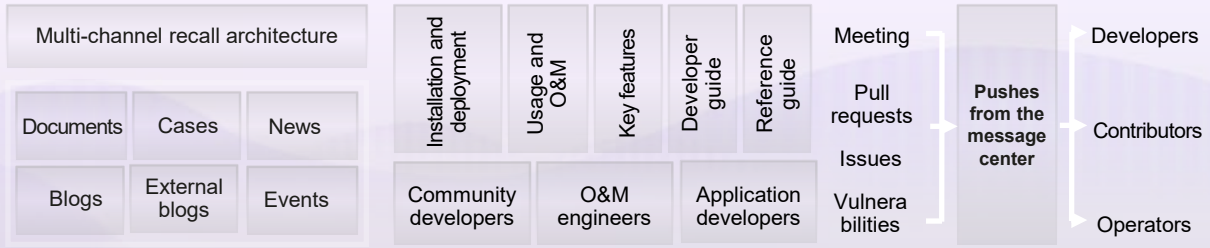
Search accuracy improved by **43.27%**
Covering mainstream channels

Upgraded documentation experience

Documentation architecture reconstruction: **5 major scenarios** and **3 types of roles**

New message center

Aggregating multi-source messages
Quick push within **1 minute**

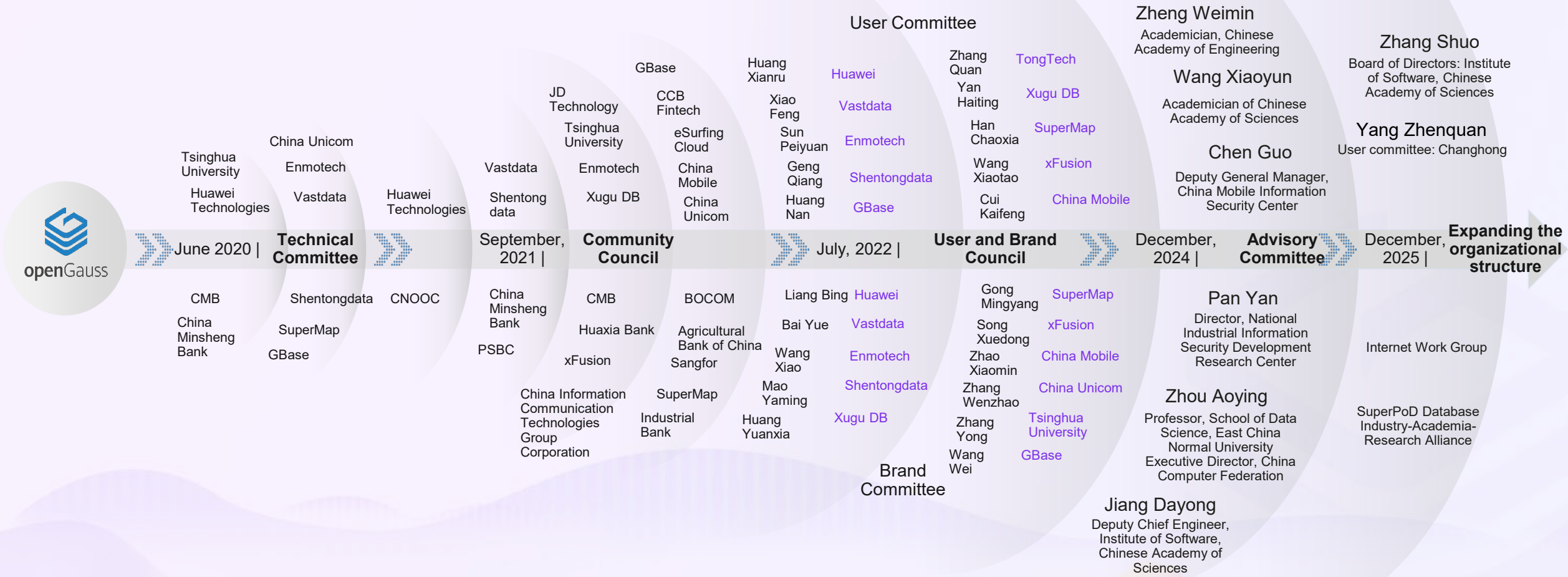


Key Community Contributions

Enhancements to advanced packages, AGG functions, Bloom index mobility, JSON operators, and SHOW syntax	Plug-ins for DataKit instance monitoring and O&M diagnostics
MySQL compatibility, CM two-node deployment, and CM support for dynamic adjustment of maximum availability mode	Audit log enhancements, OM upgrade reliability enhancements, parallel logical import/export, and logical parsing of backup files
Observability enhancements (support for querying WAL advancement information and network access whitelist information)	Inherited tables, Chinese logs, large objects, and MySQL protocol compatibility
Logical replication enhancements, resource pooling enhancements, segment-page enhancements, sorting performance optimization, and NestLoop performance optimization	...

openGauss

A Complete and Efficient Organizational Structure



Community Operations: Tiered and Multi-dimensional Community Events to Build a openGauss Tech Ecosystem

WeChat official account: https://mp.weixin.qq.com/s/dp1l3pGmHzKV04bc-HTwZA

DevOps Summit

openGauss Summit 2025

openGauss Developer Day 2025

openGauss Summit 2024

openGauss Developer Day 2024

openGauss Summit 2023

openGauss Developer Day 2023

openGauss Summit 2022

...

Face-to-face meeting

Beijing Meetup

Shenzhen Meetup

Nanjing Meetup

Hangzhou Meetup

Chongqing Meetup

Xi'an Meetup

Tianjin Meetup

Shanghai Meetup

Guangzhou Meetup

Hefei Meetup

Lanzhou Meetup

Chengdu Meetup

Changsha Meetup

Harbin Meetup

Technical Live Streams

openGauss feature course – openGauss 5.0.0 series feature interpretation

openGauss outstanding developer live stream series

openGauss feature course – openGauss 6.0.0 series feature interpretation

openGauss resource pooling live stream series

In collaboration with partners Vastdata, Enmotech, GBase, etc., we hosted **21** offline meetups, over **120** online live streams, and **9** major summits, achieving continuous breakthroughs in the MySQL tech ecosystem.

openGauss

Gathering Global Talent to Build an openGauss Talent Ecosystem

Talent layout

Establishing the openGauss A-P-E pyramid structure
A complete set of English courses

HCIA → HCIP → HCIE

Expanding from 1 to N by cultivating golden seed instructors
Enriching the university talent ecosystem

20+ training sessions

100+ trained instructors

Driving new trends in global openGauss talent development through learning and teaching via competitions

Extensive → In-depth

100+ countries and regions
30,000+ trained professionals

20+ top HCIE professionals

openGauss talent development framework and core achievements

Courses

Textbooks

Training

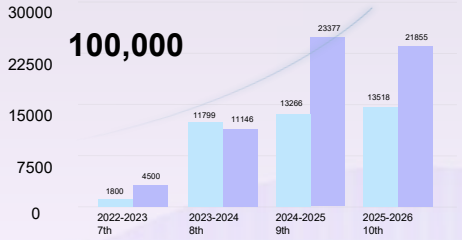
Instructor development

Universities
600+

Instructors
800+

Competitions

ICT competition



2,000+ universities in 100+ countries and regions

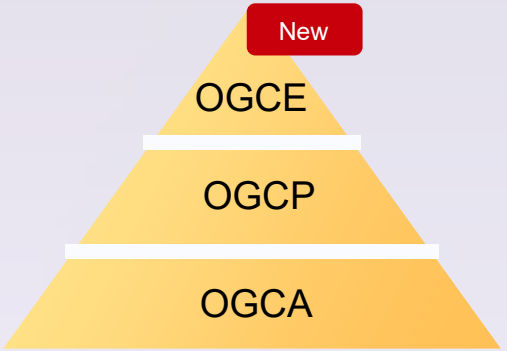
Certifications

Talent certification

Integrating national standards

DBA professionals
100,000

Community talent development



Community certification pyramid structure

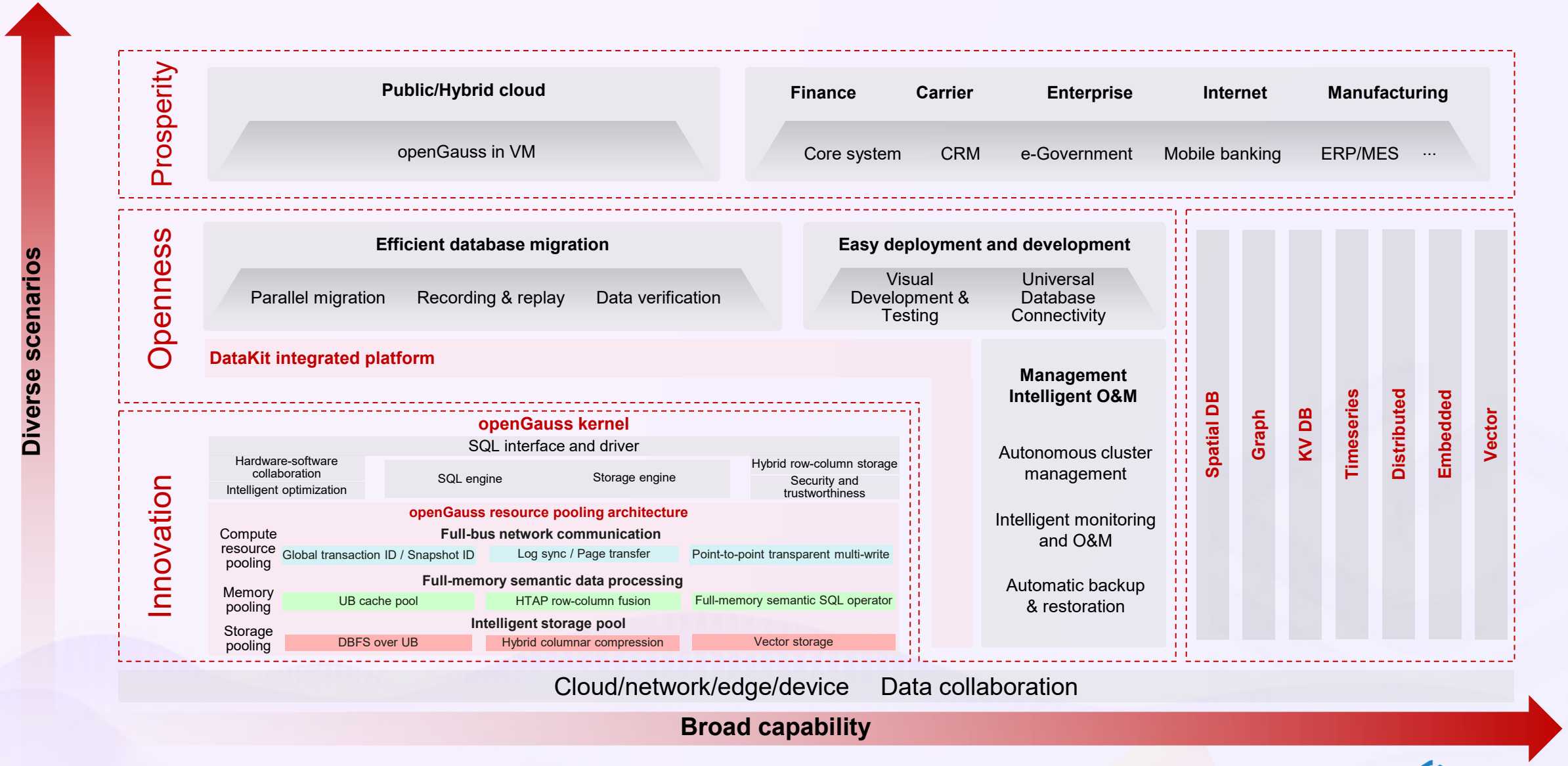
云数互联 · 海量数据 VASTDATA · 中国航天 神舟通用 · QST 青软集团 · 中国移动 China Mobile

Community certification training partners

openGauss

Directory

Technology Upgrade for an Innovative All-Scenario Database



openGauss

openGauss 7.0.0: An Open-Source Database for Digital Infrastructure

oGRAC

Multi-primary

All nodes support reads and writes with no primary/standby distinction; cross-node transaction consistency; distributed MVCC

High performance

3.5 million tpmC on two Kunpeng 920 nodes

Storage engine

Unique identifier for visibility checks; page-level MVCC; transaction-slot-based row locking

Kernel upgrades

Parameterized path

Automatic parameterization of common INSERT/UPDATE/DELETE/SELECT statements, improving performance by 10%+

Online DDL

Significantly reduces high-level lock holding time during DDL, minimizing blocking of concurrent DML

DPA hash aggregation acceleration

Offloads vectorized hash aggregation to hardware accelerators, improving performance by over 20% in complex AP scenarios

Async direct I/O

Data access uses direct I/O to bypass the OS page cache; page flushing supports asynchronous I/O

Black box

Protects database write-ahead logs (WAL) to build a secure anti-ransomware solution

Fully enhanced AI capabilities

DataVec: Combined with Kunpeng RAG appliance, uses quantization acceleration and vector-scalar fusion to mitigate knowledge hallucinations in LLMs such as DeepSeek, improving response timeliness and accuracy

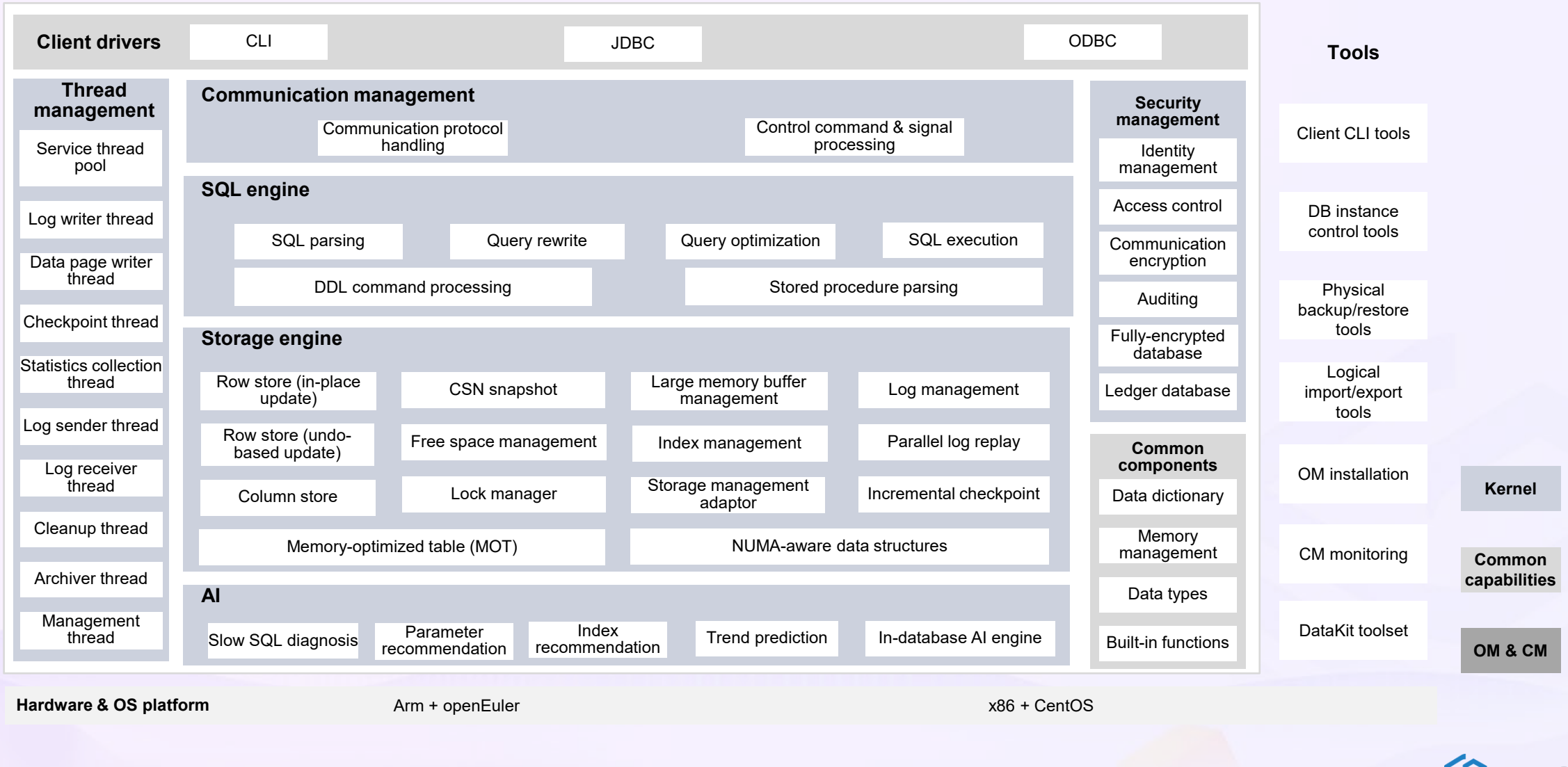
Supports oGAI, a one-stop AI capability integration solution that allows AI models to be invoked directly within the database for text vectorization, text generation, document re-ranking, and more, without relying on the external application layer

Newly supports BM25 full-text indexes, DiskANN disk-based indexes, and RaBitQ quantization indexes

Adapts to MCP, enabling AI agents to securely and efficiently operate databases through standardized protocols

Directory

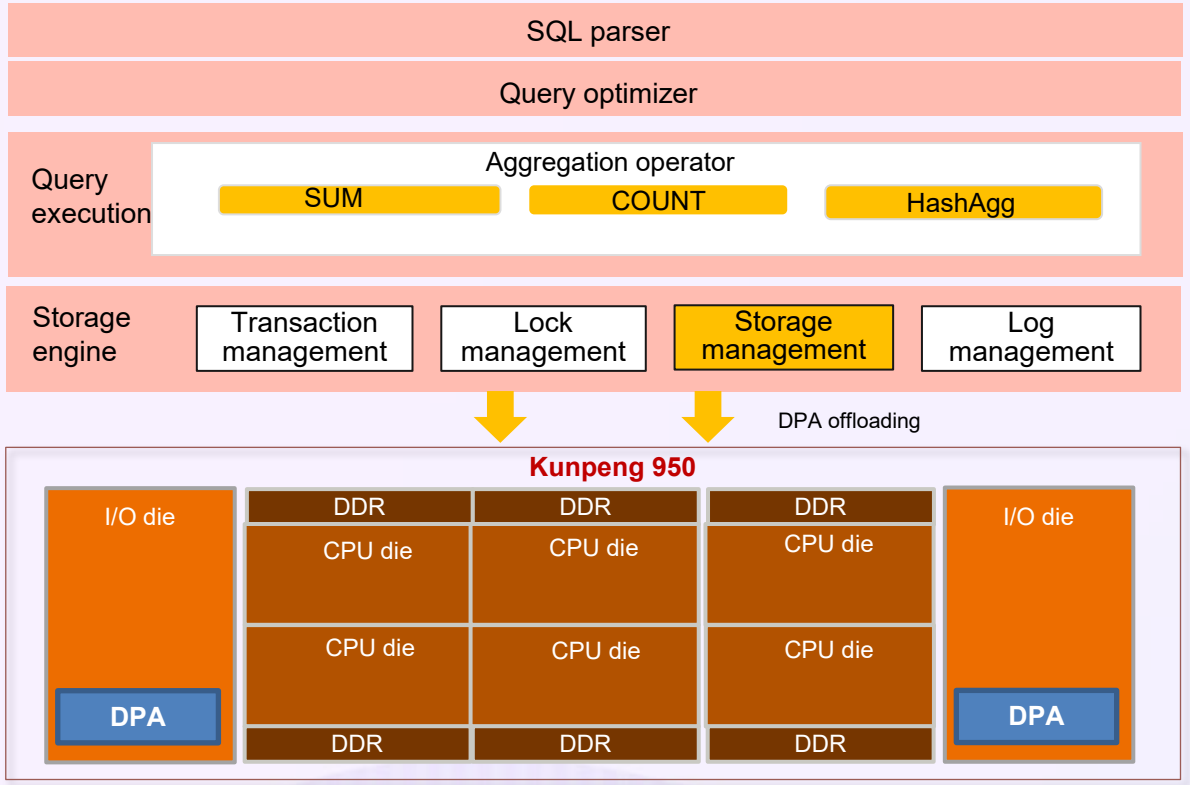
openGauss Logical Modules



openGauss

High Performance: DPA Hash Aggregation Acceleration

In large-scale data aggregation queries, the system offloads hash aggregation from the CPU to a hardware accelerator, reducing CPU overhead and improving query performance.



Key tech

- **Aggregation offloading: Offloads SUM/COUNT etc. to DPA, reducing CPU load**

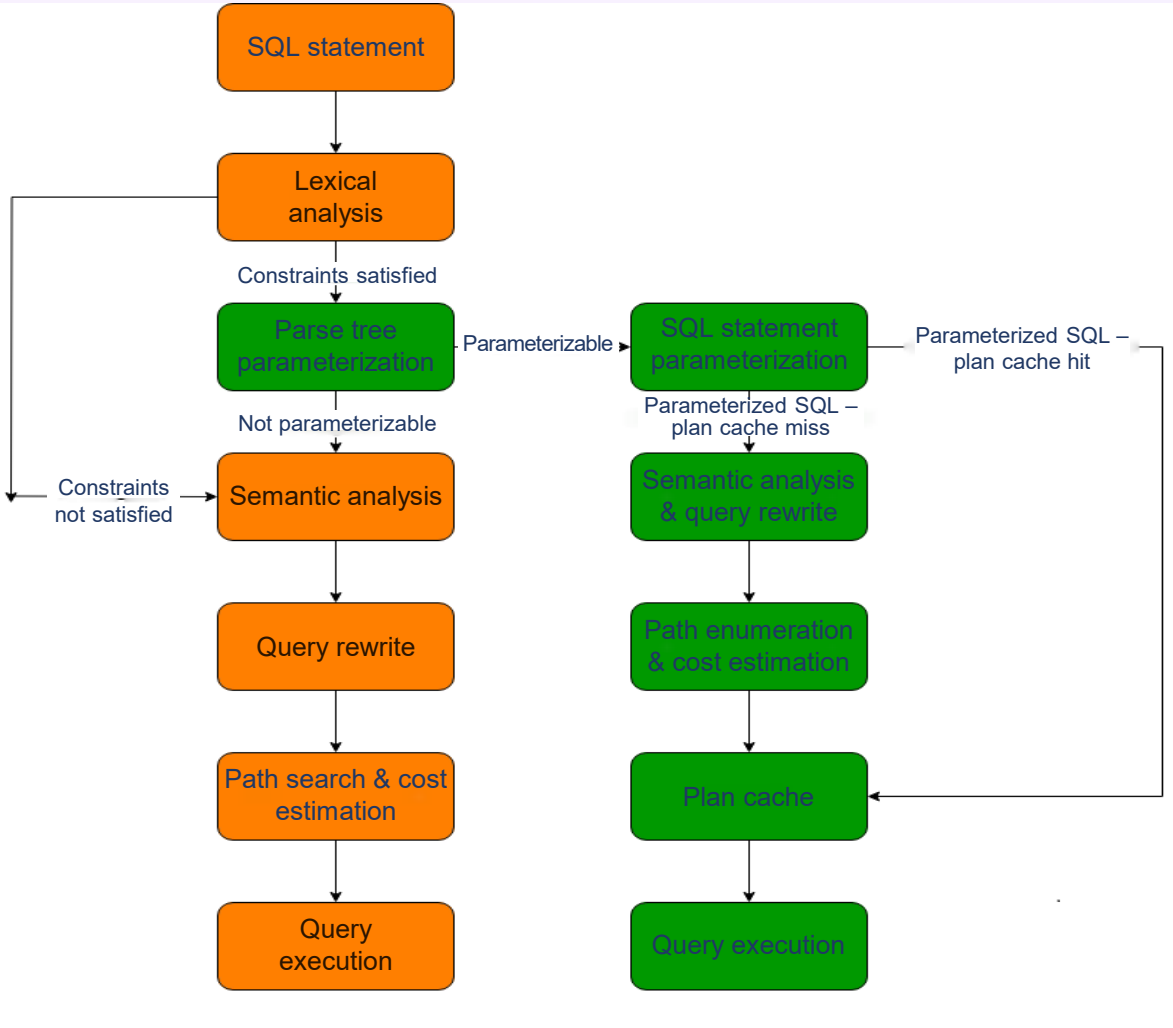
Benefits

- **Complex analytical aggregation queries: 20%+ performance improvement**

openGauss

High Performance: Parameterized Paths with Automatic Parameterization & Plan Caching

SQL statements are parameterized and used as cache keys, with their corresponding execution plans cached. Subsequent executions of the same parameterized SQL template can directly hit the plan cache, reducing the time spent on query rewrite and plan generation.



Key tech

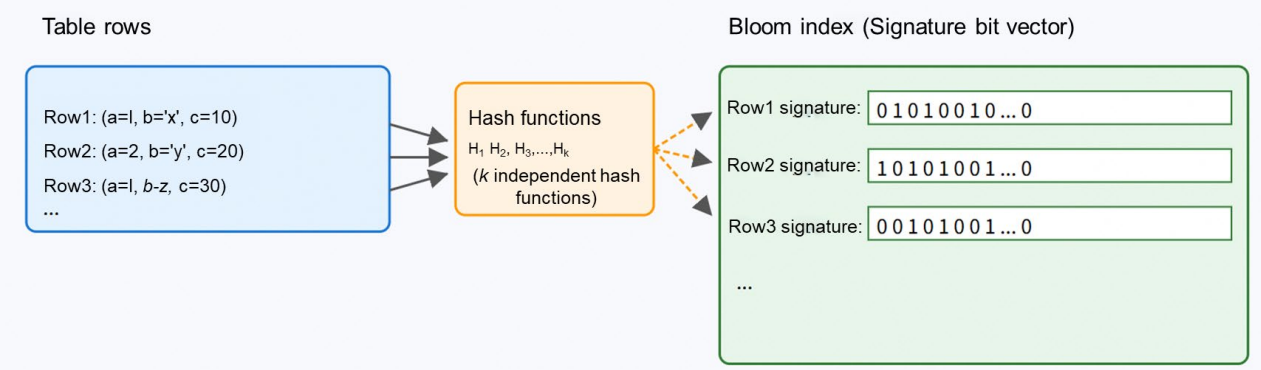
- **Auto-parameterization:** Identifies and extracts constants from common SQL statements, parameterizes them, and generates parameterized execution plans.
- **Plan caching & reuse:** The same parameterized SQL template can reuse an existing cached execution plan, reducing plan generation overhead.

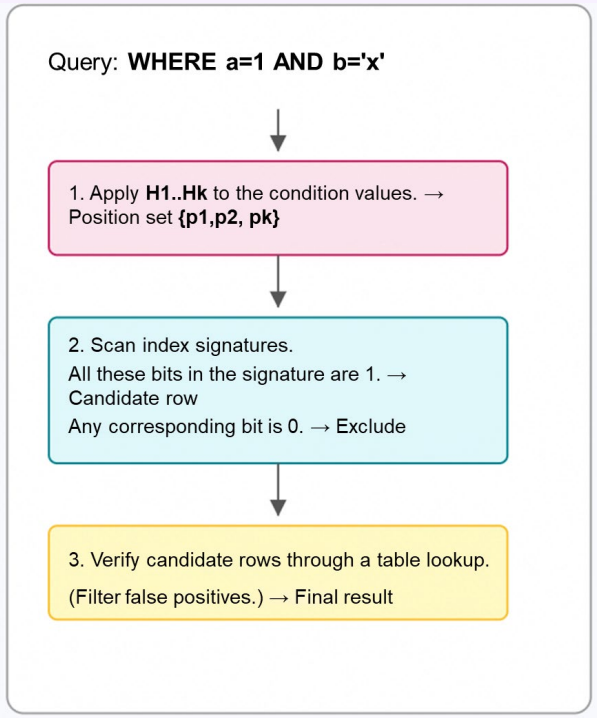
Benefits

- Repeated execution of parameterized SQL improves performance by over 20%.

openGauss

High Performance: Bloom Index Balances Storage and Performance





Key tech

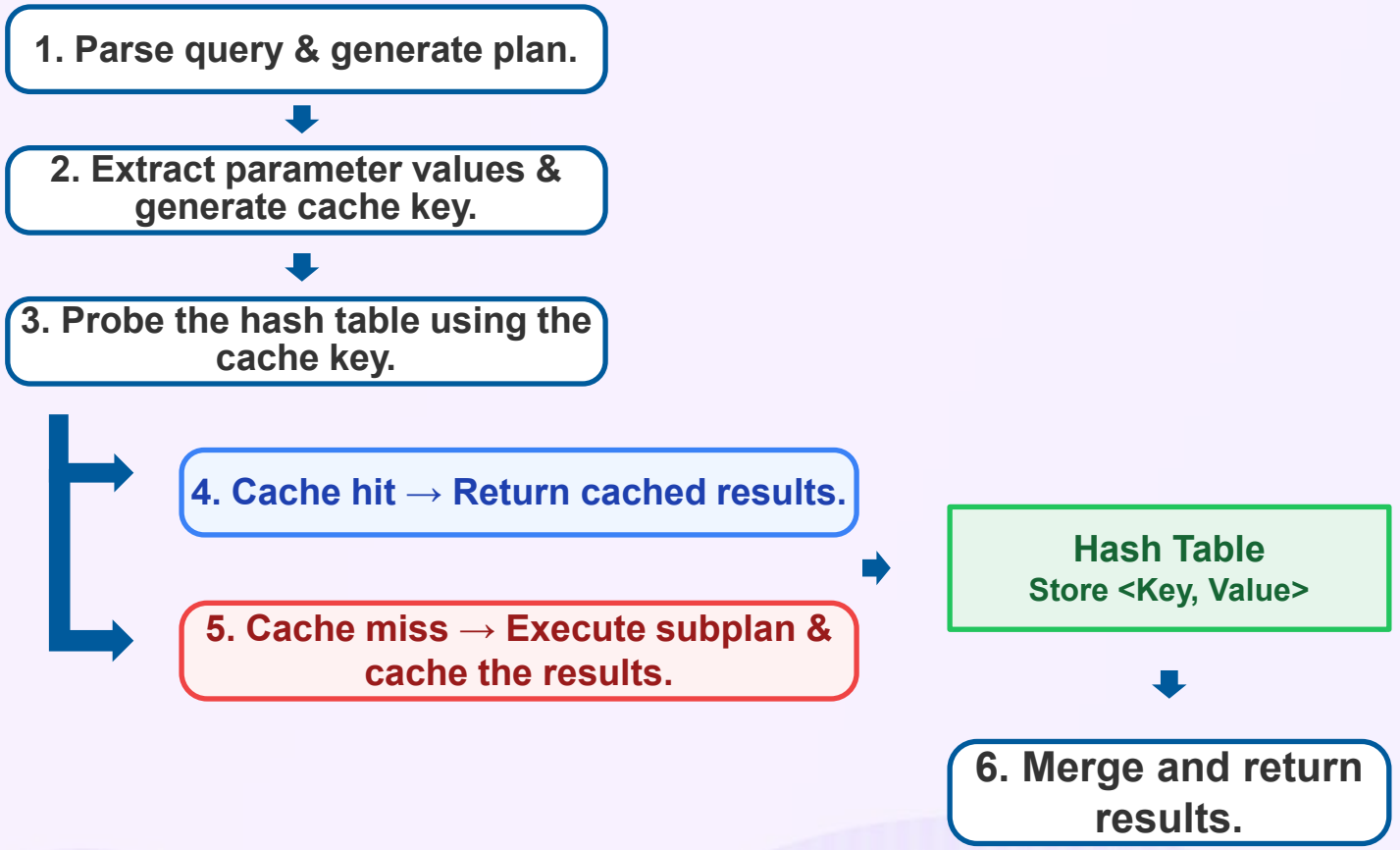
- **Accelerated multi-column equality queries:** Bloom indexes are designed specifically for the **=** operator, ideal for multi-column queries, such as **WHERE col1 = v1 AND col2 = v2 AND col3 = v3**.
- **Signature (bitmap) storage:** Each row is hashed into a fixed-length bit vector (signature). The index stores only signatures rather than actual column values, requiring much less space than a B-tree composite index.
- **False positives, no false negatives:** The query signature is compared with the index signature using a bitwise AND operation. If all required bits are 1, the row may match and requires a table lookup. If any required bit is 0, the row is guaranteed not to match. This allows the index to quickly eliminate non-matching rows.

Benefits

- **Improved query performance:** Bloom indexes reduce table lookups for multi-column equality queries and outperform B-tree composite indexes when column selectivity is low (for example, many duplicate values).
- **High space efficiency:** Compared with B-tree composite indexes, Bloom indexes store only compact signatures, reducing drive space by over 50%.

openGauss

High Performance: New Memoize Operator Optimizes Nested Loop Performance



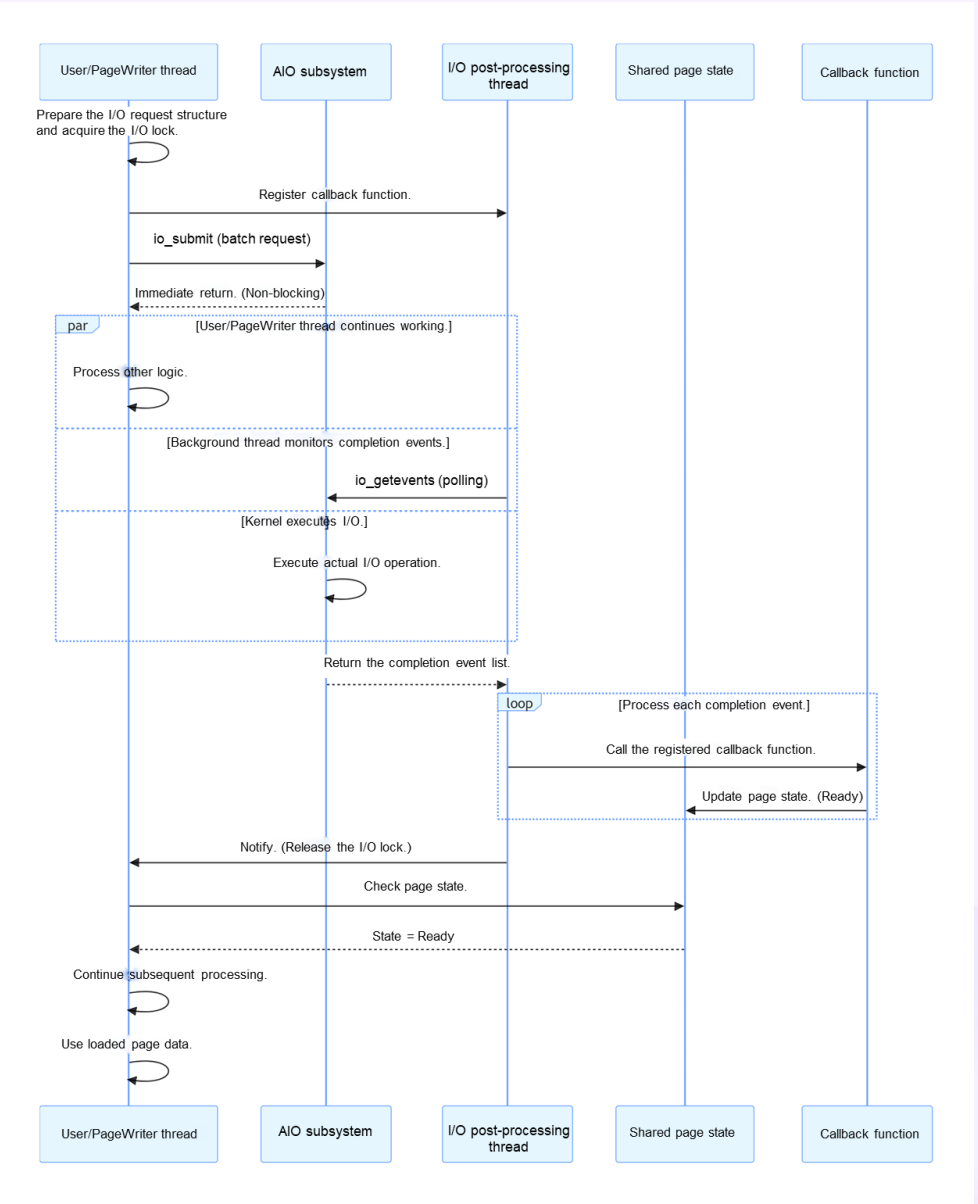
Key tech

- **Result set caching:** Stores subplan results in memory for fast access.
- **Parameterized cache key**: Built from subplan parameterized columns for precise cache matching.

Benefits

- 100%+ faster for Nested Loop joins with repeated outer table scans.

High Performance: Async Direct I/O



Key tech

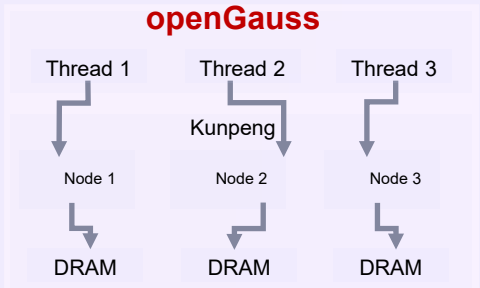
- **I/O request submission:** The user thread builds an I/O request acquires an I/O lock on the page to prevent concurrent conflicts. AIO returns immediately, keeping the user thread non-blocking.
- **Three-way parallel async execution**: The user thread continues processing other tasks, while a background thread polls the kernel for I/O state and the kernel AIO subsystem handles drive I/O asynchronously.

Benefits

- 10%+ faster write-only workloads on low-performance drives.

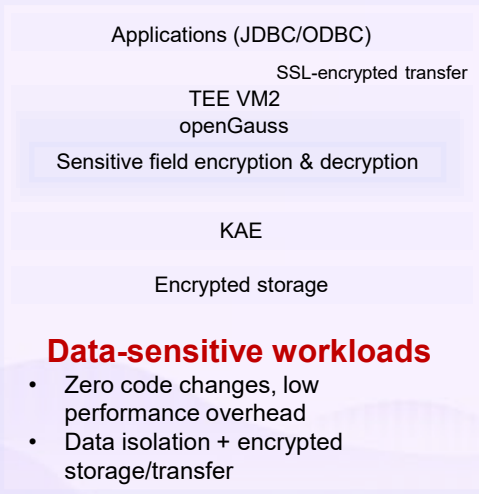
openGauss

High Performance: openGauss + Kunpeng Unleash Superior Computing Power



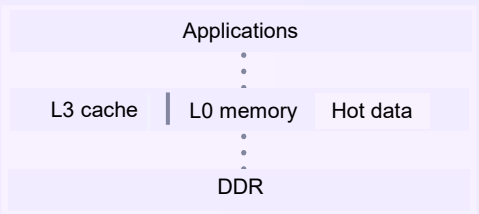
TP/AP workloads

NUMA-aware global core data structure redesign and thread pinning to improve performance



Data-sensitive workloads

- Zero code changes, low performance overhead
- Data isolation + encrypted storage/transfer



Data-intensive workloads

Hot data and locks stored in high-performance L0 memory for faster access

Pain points

- Application threads randomly scheduled across NUMA cores, causing high CPU overhead.
- High encryption overhead in data-sensitive scenarios.
- Inefficient memory access for hot data in data-intensive scenarios.

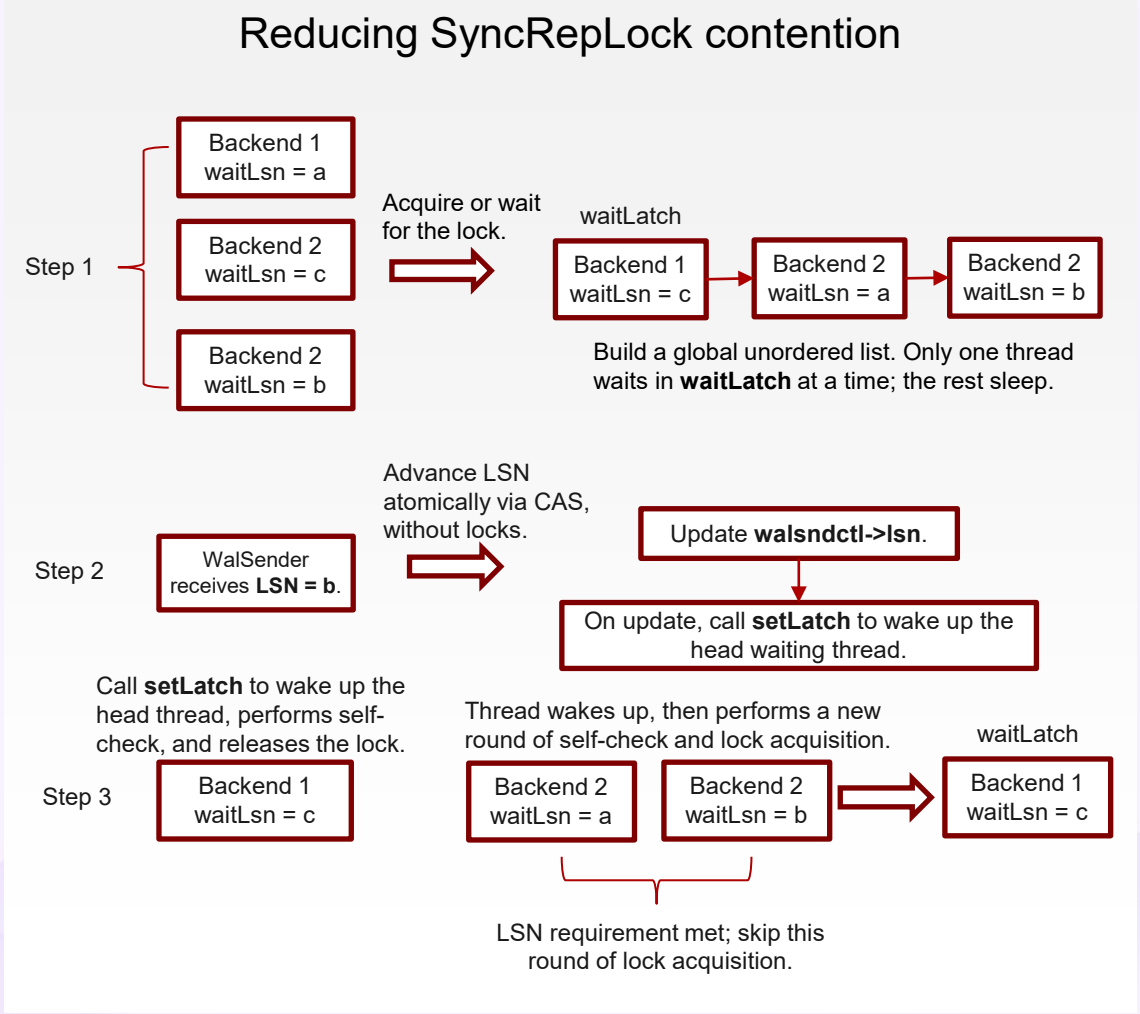
Key tech

- **Thread pinning**: Separately pins network interrupts, background threads, and application threads to dedicated cores; apply NUMA-aware hot data structures to reduce cross-core access.
- **Kunpeng Accelerator Engine (KAE)**: Hardware-accelerated encryption with zero code changes.
- **High-speed L0 memory**: Stores hot data in L0 memory.

Benefits

- Based on Kunpeng 920, the TPCC single-node performance reaches 1.5 million tpmC.
- KAE hardware encryption enables performance loss under 20%.
- Hot data and locks stored in L0 memory for 10% faster access.

High Performance: SyncRepLock Refactoring Enables Fine-Grained Locking



Pain points

- SyncRepLock controls both primary/standby state changes and SyncRepQueue operations, leading to severe lock contention.
- WalSender handles multiple tasks, including waking up queues and updating LSN values.

Key tech

- **SyncRepLock refactoring**: Controls only primary/standby state changes, no longer managing SyncRepQueue.
- **Lock-free queue**: Refactors SyncRepQueue into multiple unordered lock-free lists.
- **Atomic operations**: Updates LSN values atomically via CAS without locks.

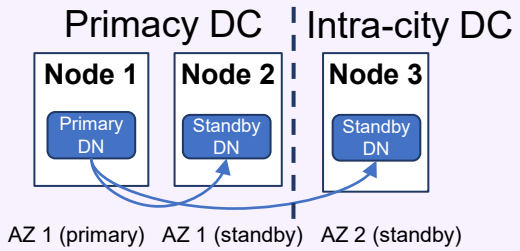
Benefits

- With one primary and one synchronous standby, **TPC-C performance improves by over 10%, reaching 1.3 million+ tpmC.**

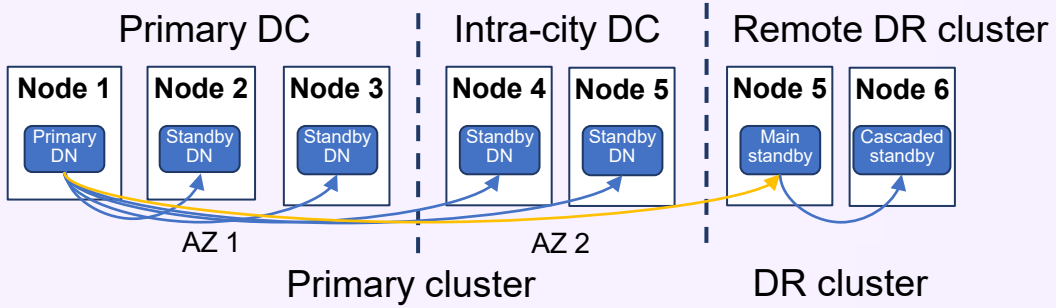
openGauss

Supporting Multiple Networking Modes to Match Diverse HA Requirements

Primary/Standby deployment

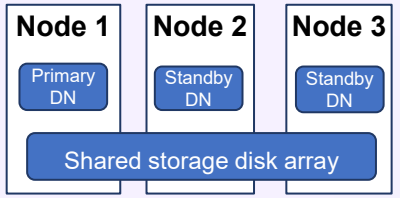


Dual-DC intra-city deployment:
1 primary node and *X* standby nodes (1+*X* servers)
RPO = 0, RTO < 10s

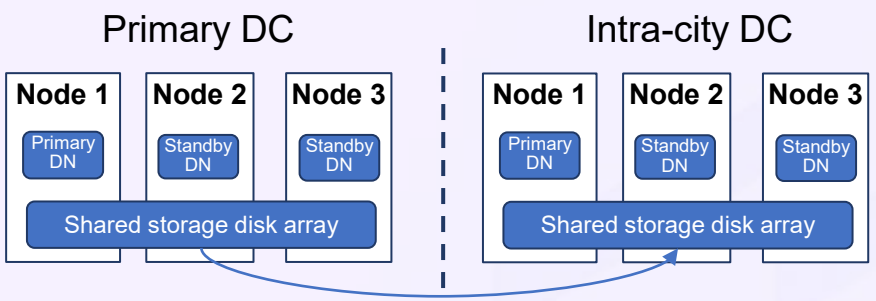


Geo-redundant 3DC DR deployment:
1 primary node and *X* standby nodes (*X* ≥ 3) (1+*X* servers)
Intra-city synchronous replication with RPO = 0, remote asynchronous replication DR with RPO ≈ 0
Failover to DR DC, RTO < 10 mins

DataPod deployment



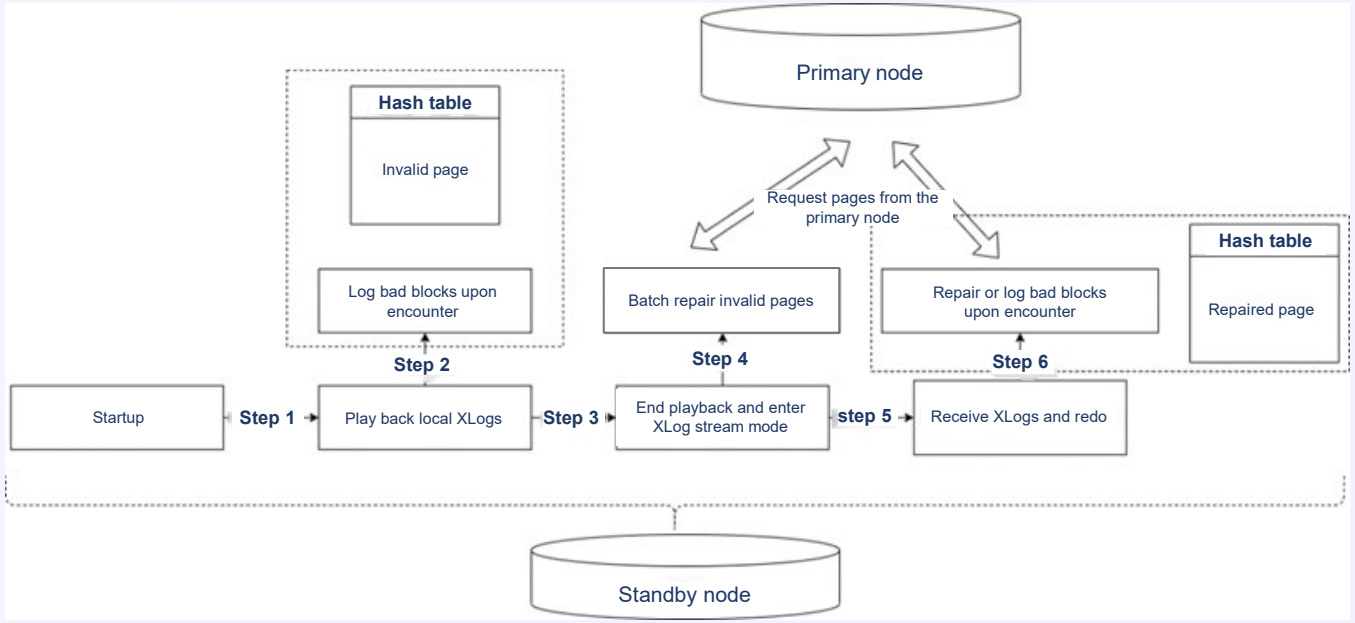
Single-DC deployment:
HA deployment based on the primary/standby resource pooling of disk arrays
1 primary node and *X* standby nodes (1+*X* servers)
RPO = 0, RTO < 10s



Dual-DC intra-city deployment:
Dual-cluster intra-city DR based on storage synchronous replication
1 primary node and *X* standby nodes (*X* ≥ 3) (1+*X* servers)
RPO = 0, RTO < 30s

	Memory	CPU	Storage	Network
Single server configuration (commercial deployment)	Recommended: ≥ 128 GB	Recommended: 1 × 16 cores, ≥ 2.0 GHz	Primary/standby: Minimum 20 GB required, with at least 70% of remaining disk space reserved for data storage. SSDs are recommended as the primary storage. DataPod: Minimum 10 GB required for data storage and 10 GB required for log storage. An OceanStor Dorado disk array is required, with a single center sharing one disk array.	Recommended: ≥ 300 Mbit/s Ethernet NIC

High Availability: Bad Block Healing via Cross-Node Repair



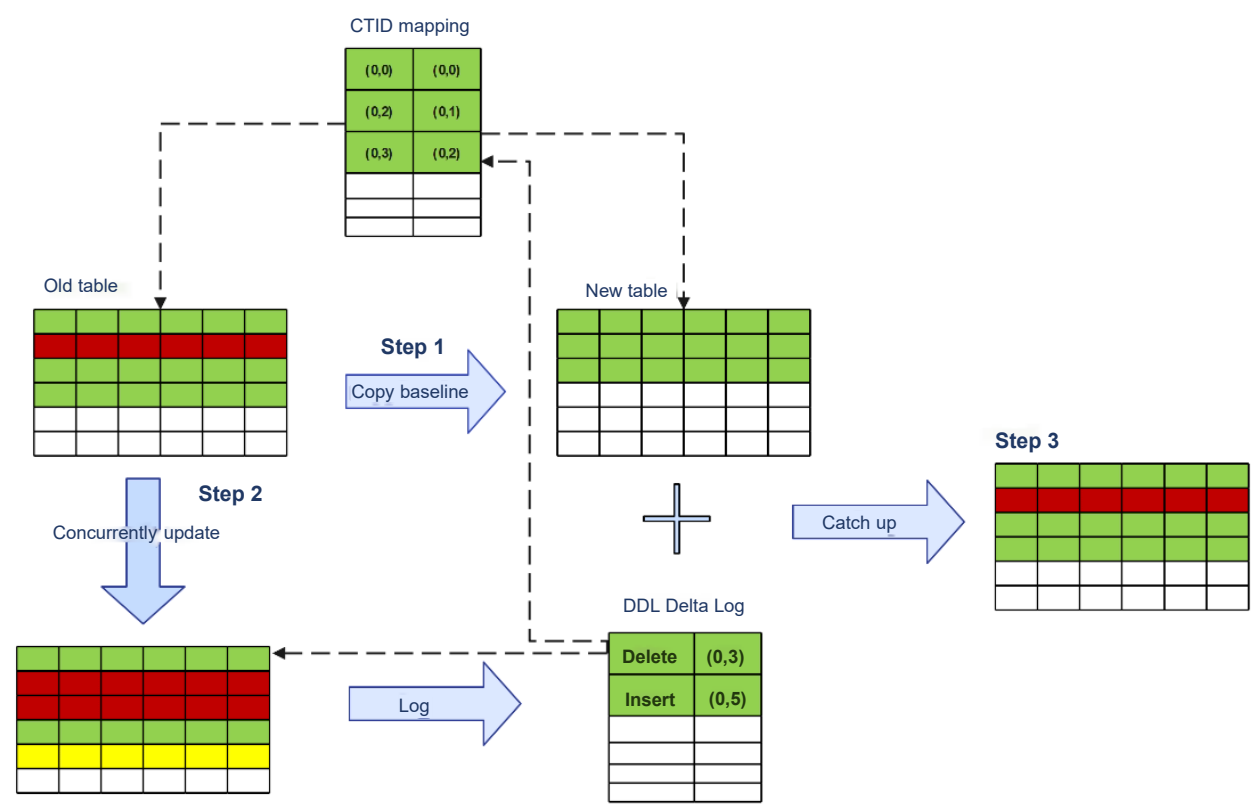
Key tech

- **Standby-to-primary repair**: When the primary node detects a corrupted page during a read, the page read function remotely connects to the standby node to retrieve the normal page, and returns it to the upper layers. The bad block remains completely transparent to the user.
- **Primary-to-standby repair**: If the standby node detects a corrupted page, it retrieves the correct data from the primary node and overwrites the underlying page.

Benefits

- **Disk bad block repair**: When disk data is corrupted on either the primary or standby node, the system recovers by pulling correct data from other nodes, ensuring database availability.

High Availability: Minimizing DML Impact with Online DDL



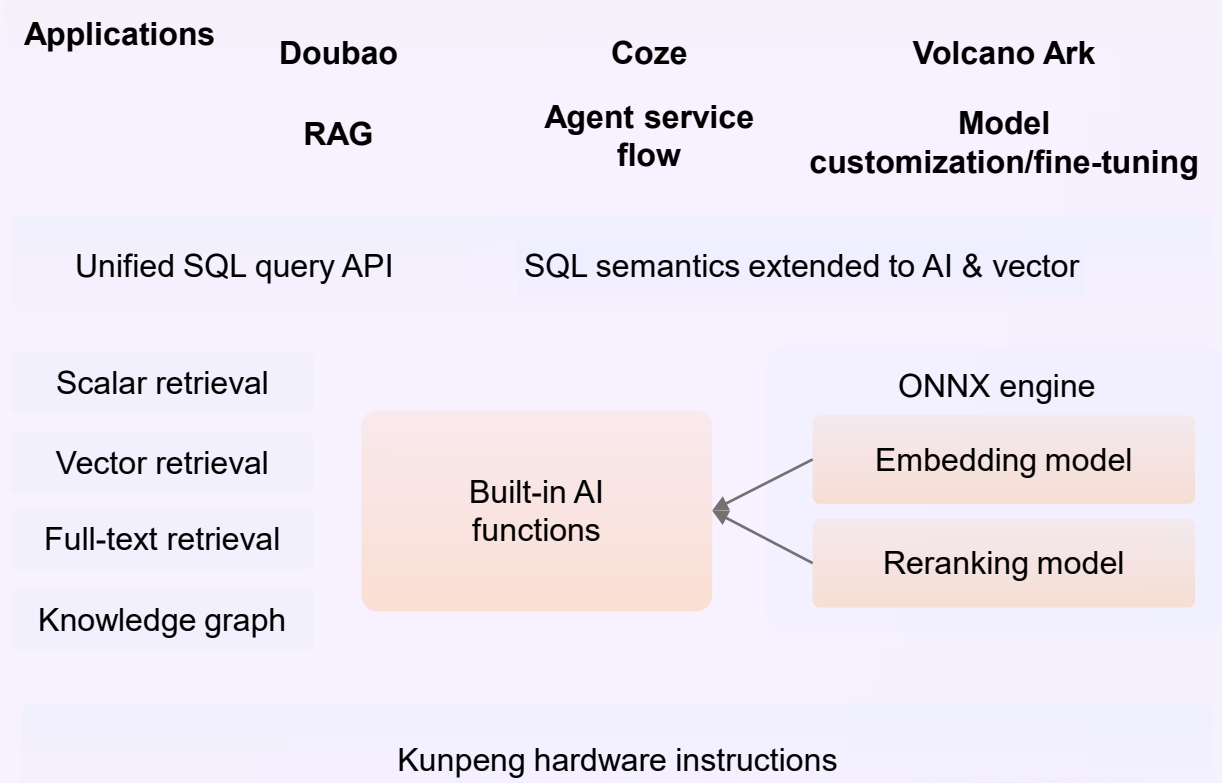
Key tech

- **Baseline table + Delta table for change capture:** The old table is cloned to the new table created based on the new metadata. During online DDL, the IUD operations on the old table are recorded in the Delta table.
- **CTID mapping table**: The CTID mapping between the old and new tables is recorded in the CTID table.

Benefits

- Supports online execution for most DDL, VACUUM FULL, and CLUSTER operations, reducing the duration of holding high-level locks by 90%+.

High Intelligence: One-Stop AI Capability Integration with AI Pipeline



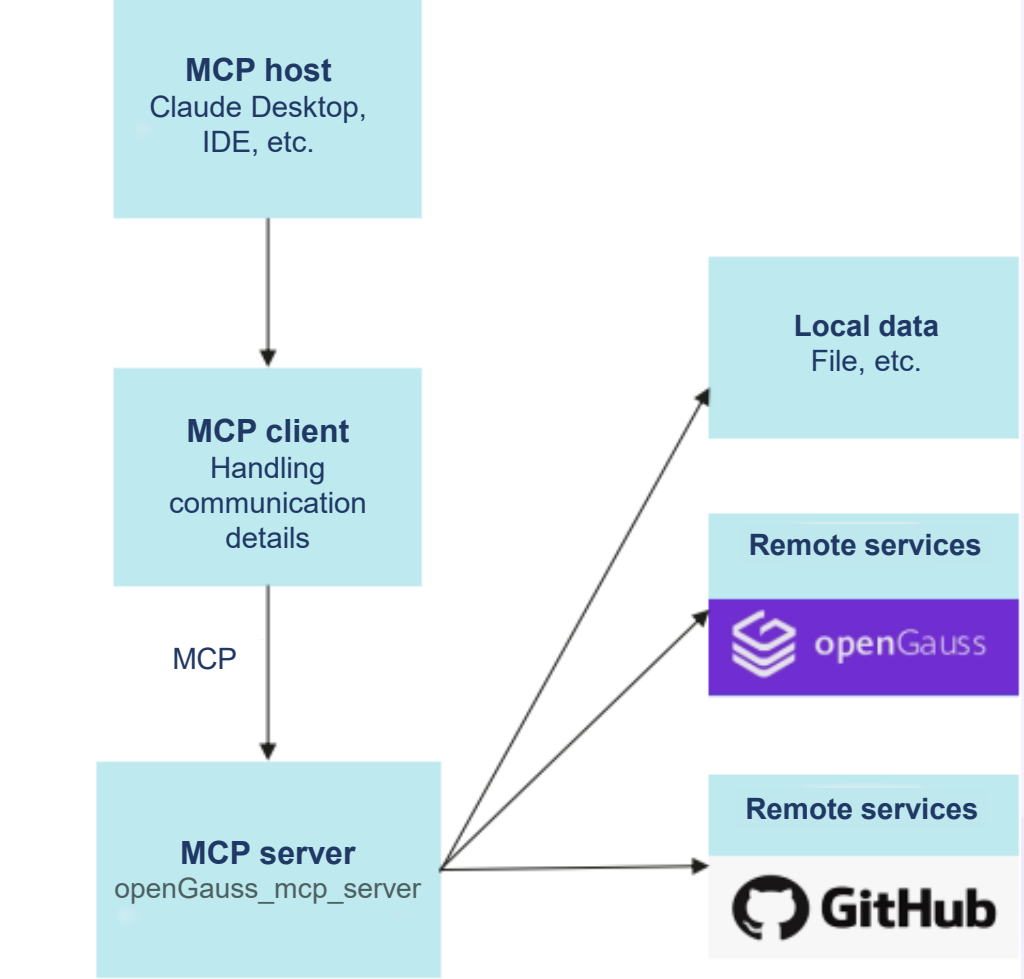
Key tech

- **In-kernel HTTP requests to external models**: Accesses external models via **libcurl**, providing Embed, Chat, and Rerank APIs.
- **Extended SQL semantics**: Enables automated data vectorization to reduce data copying. Incremental data is automatically vectorized, providing continuous, AI-powered QA enhancement.

Benefits

- **ONNX model deployment**: Built-in lightweight ONNX models ensure data privacy within the local trust zone, boosting data security.
- **One-stop RAG QA:** SQL UDFs can be directly invoked to implement knowledge QA capabilities in natural language.

High Intelligence: MCP for openGauss, Empowering AI Agents with Standardized Database Operations



Key tech

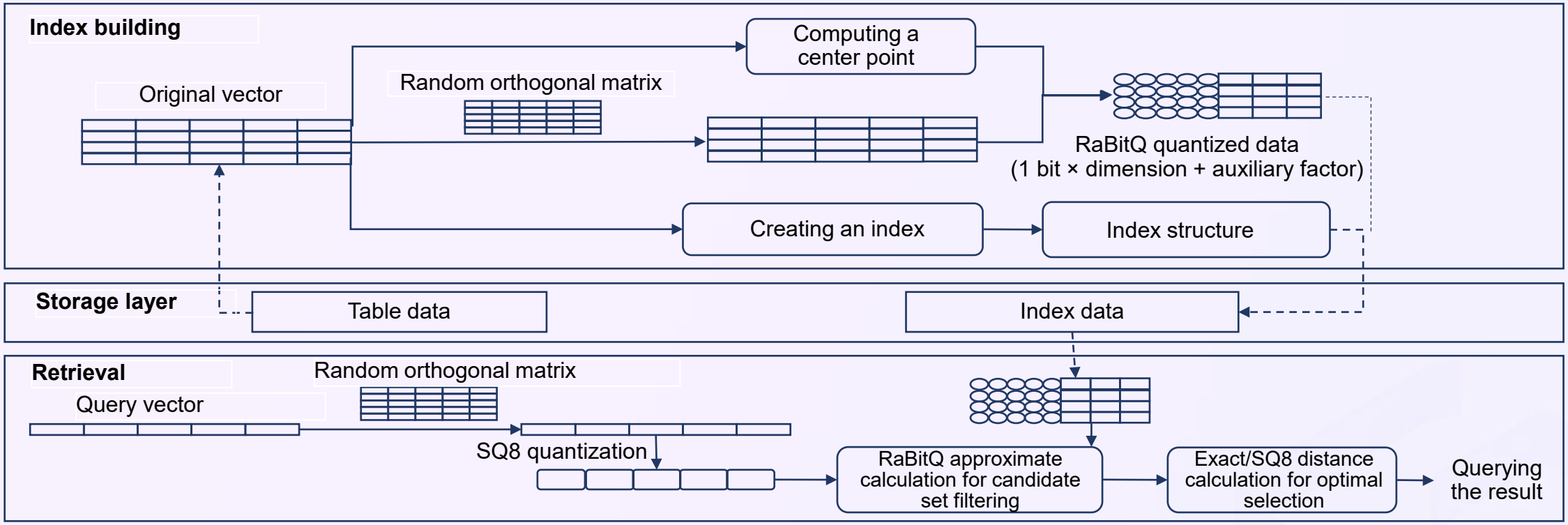
- **openGauss MCP server:** Implements multi-tool capabilities, enabling standardized calling by any MCP host.

Benefits

- **Rich MCP APIs**:
  - SQL execution, table discovery, data preview, and execution plan retrieval
  - BM25 full-text index creation, hybrid search, and official documentation access
  - User memory system

openGauss

High Intelligence: RaBitQ Algorithm for High Compression with Uncompromised Search Reliability



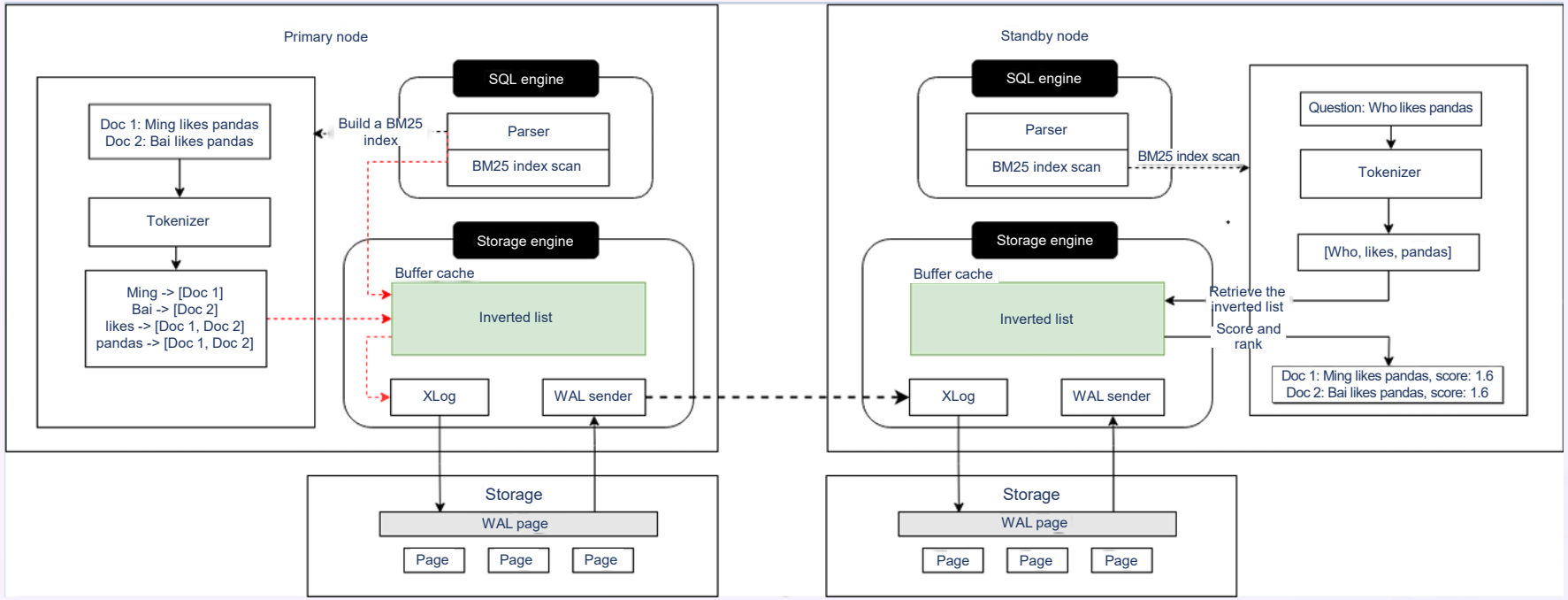
Key tech

- **Normalized data vector**: Maps both data and query vectors onto a unit hypersphere, eliminating distribution skew and simplifying distance calculations into inner products.
- **SQ8 quantization**: Compresses query vectors via SQ8, representing each dimension with 8 bits.

Benefits

- **RaBitQ for IVF/HNSW indexes**: Delivers a 4x reduction in memory consumption compared to standard IVF/HNSW indexes.

High Intelligence: BM25 Full-Text Search for Rapid Document retrieval



Key tech

- **Tokenizer**: Introduces the open-source Jieba tokenizer to segment documents and build inverted indexes.
- **Search**: Tokenizes user queries, retrieves relevant inverted lists based on the tokens, scores and ranks documents via the BM25 algorithm, and returns the top-K highest-scoring documents to the LLM for generating answers.

Benefits

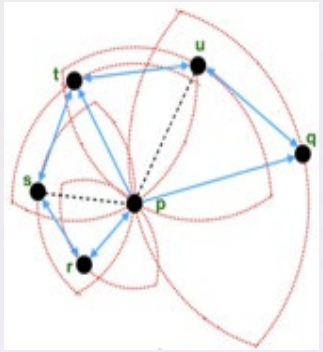
- The query response performance is more than 10 times that of GIN indexes.

openGauss

High Intelligence: DiskANN Support for Faster Document Retrieval

DiskANN storage architecture

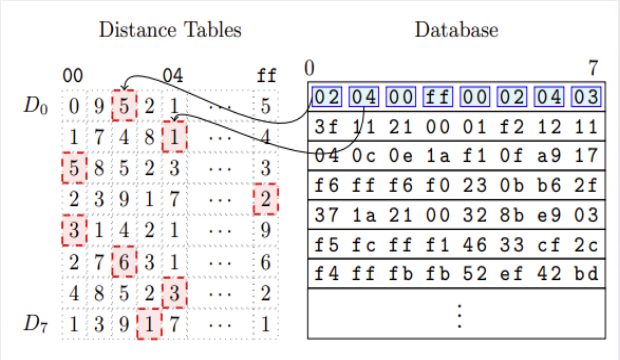
QSG edge selection and neighbor adjustment



Reduced comparisons through precise edge selection, graph degree control, and optimized neighbor selection

DiskANN memory-array disk storage architecture

Optimal PQ distance calculation acceleration algorithm



Accelerated distance calculation through PQ register adaptation, optimized PQ code layout, and reduced lookup table loading

Key tech

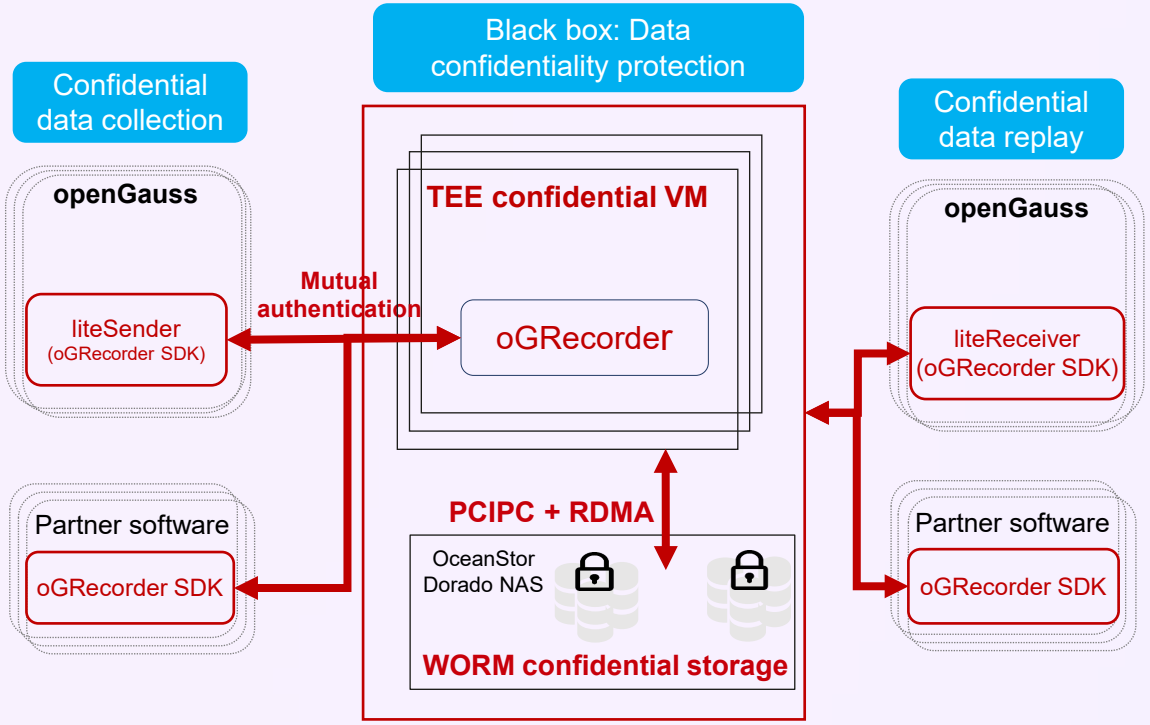
- **Vamana graph index**: PQ compression for rapid search narrowing
- **Storage optimization**: Access frequency-aware data placement to reduce I/O
- **High accuracy**: Approximate search with exact calculation for high retrieval quality

Benefits

- On the Cohere 1M benchmark, DiskANN delivers 10%–30% higher QPS than HNSW at the same recall rate.

openGauss

High Security: Data Vault for WAL Integrity Protection with oGRecorder



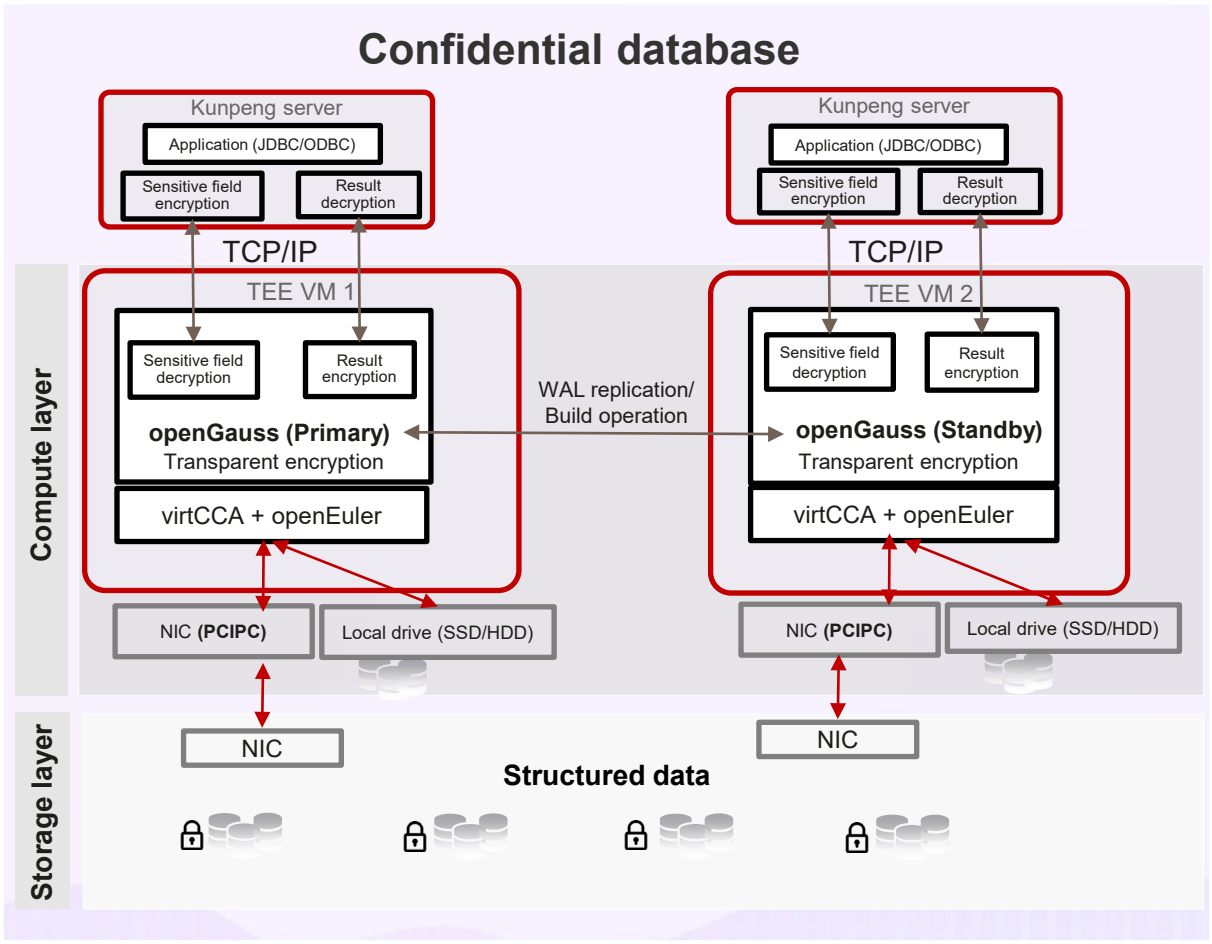
Key tech

- **oGRecorder**: Provides mutual authentication for secure connections with the oGRecorder SDK, data tamper-proof verification, and real-time log synchronization.
- **TEE confidential computing**: Provides a hardware-isolated TEE environment for oGRecorder, isolated from the host OS and REE.
- **Confidential storage**: Append-only storage with WORM support to prevent data modification and deletion, ensuring immutable backup data protection.

Benefits

- Mutual authentication prevents unauthorized access; TEE-protected VMs defend against hijacking; confidential storage protects against data tampering and destruction.
- Multi-node deployment and real-time log replay ensure post-incident recovery with RPO ≈ 0.

High Security: virtCCA-based Confidential Database Solution, < 20% Performance Loss



Pain points

- Data flows through multiple stages from applications to storage, introducing potential data leakage risks at every stage. An end-to-end solution is required to provide full-pipeline data privacy protection.

Key tech

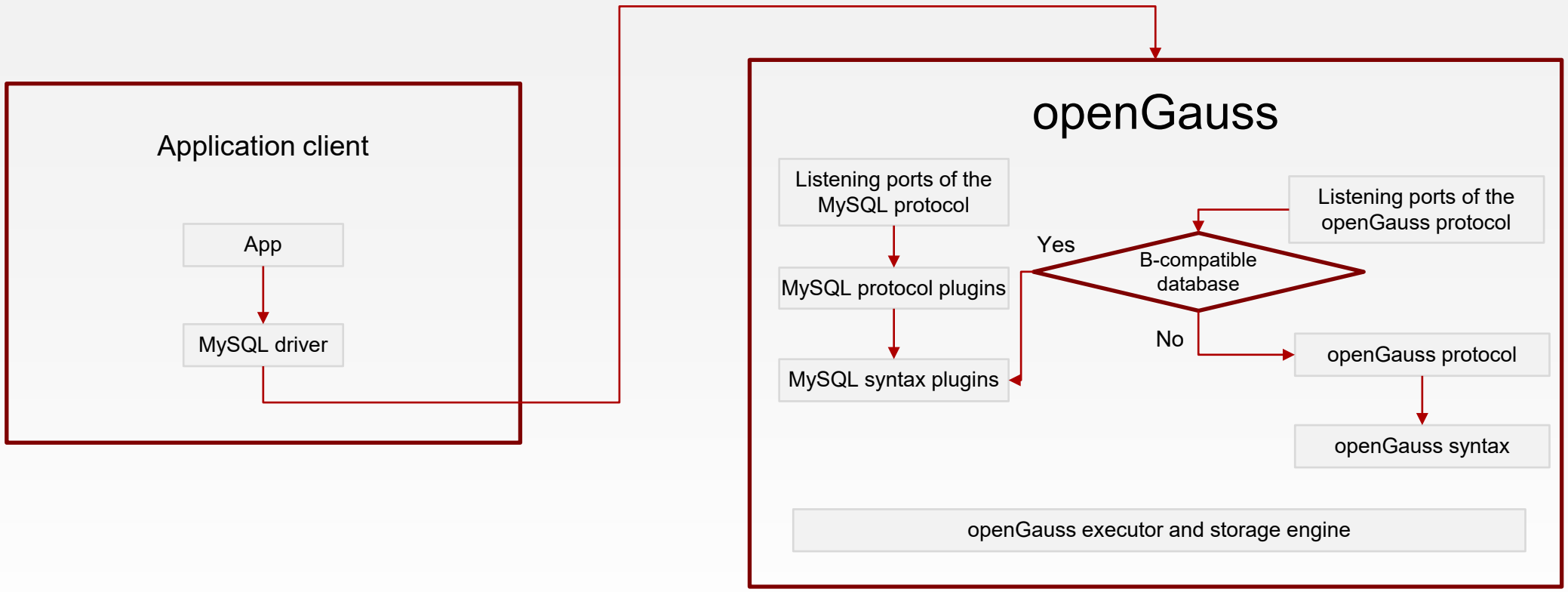
- **Confidential VM**: Isolates its memory from the host for secure computing within the VM.
- **Sensitive field encryption and decryption**: The database runs in virtCCA. Keys are stored only in the database memory and are not flushed to drives.
- **Transparent encryption**: Data is transparently encrypted and flushed to drives.

Benefits

- E2E data encryption is supported with less than 20% performance overhead compared with traditional approaches.

openGauss

High Compatibility: 100% Compatibility with Common MySQL Syntax and Protocols



Pain points

- Limited MySQL compatibility increases application migration costs and requires extensive SQL modifications.

Key tech

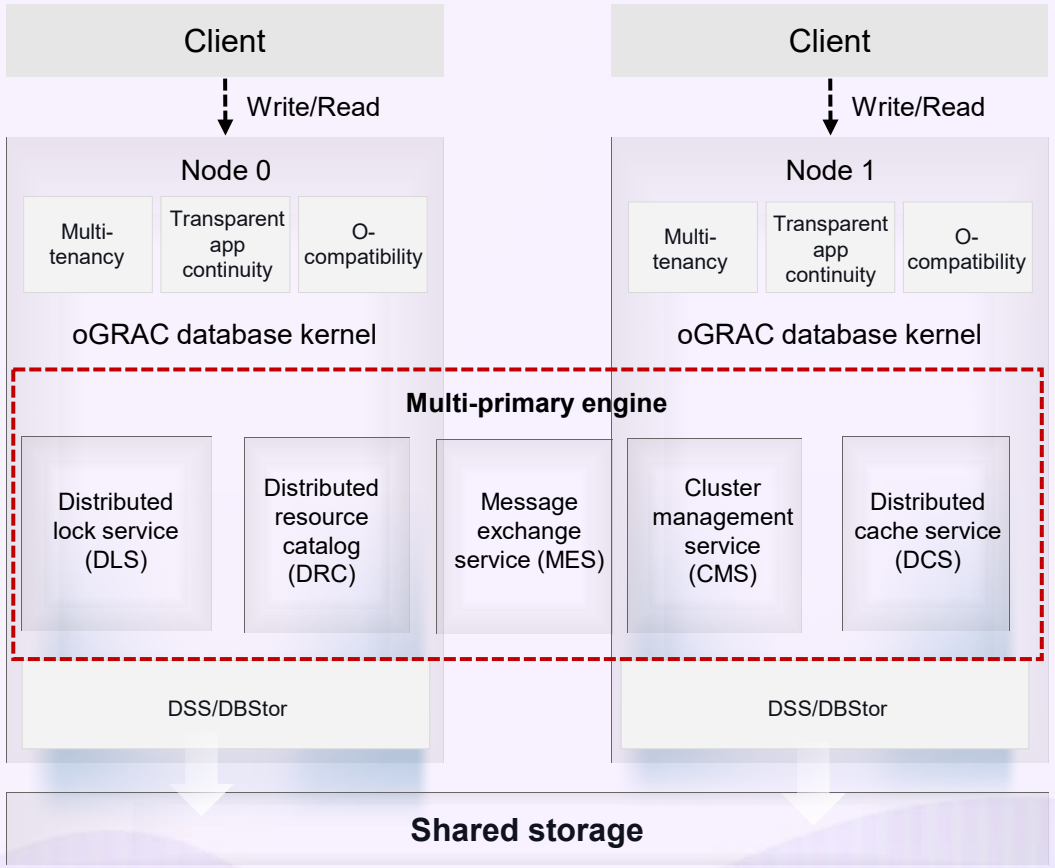
- Plugin-based architecture: Implements a dedicated MySQL syntax plugin, with all modifications isolated from the kernel.
- Protocol-level compatibility: Supports native MySQL clients and JDBC connections for seamless access to openGauss.

Benefits

- 100% compatibility with common MySQL syntax
- Compatible with common protocols of MySQL 5.7 and 8.0

Directory

oGRAC: Industry's First Open-Source Multi-Write Database



Engine-enabled multi-write, writable/readable on any node

01

Global distributed cache: ensuring cross-node transaction consistency

02

Distributed MVCC: less impact on transaction commit performance

03

Distributed lock: cross-node resource sharing and concurrency control, reducing read-write conflicts and enabling high-performance concurrent writes

Precise detection and fast recovery of cluster faults

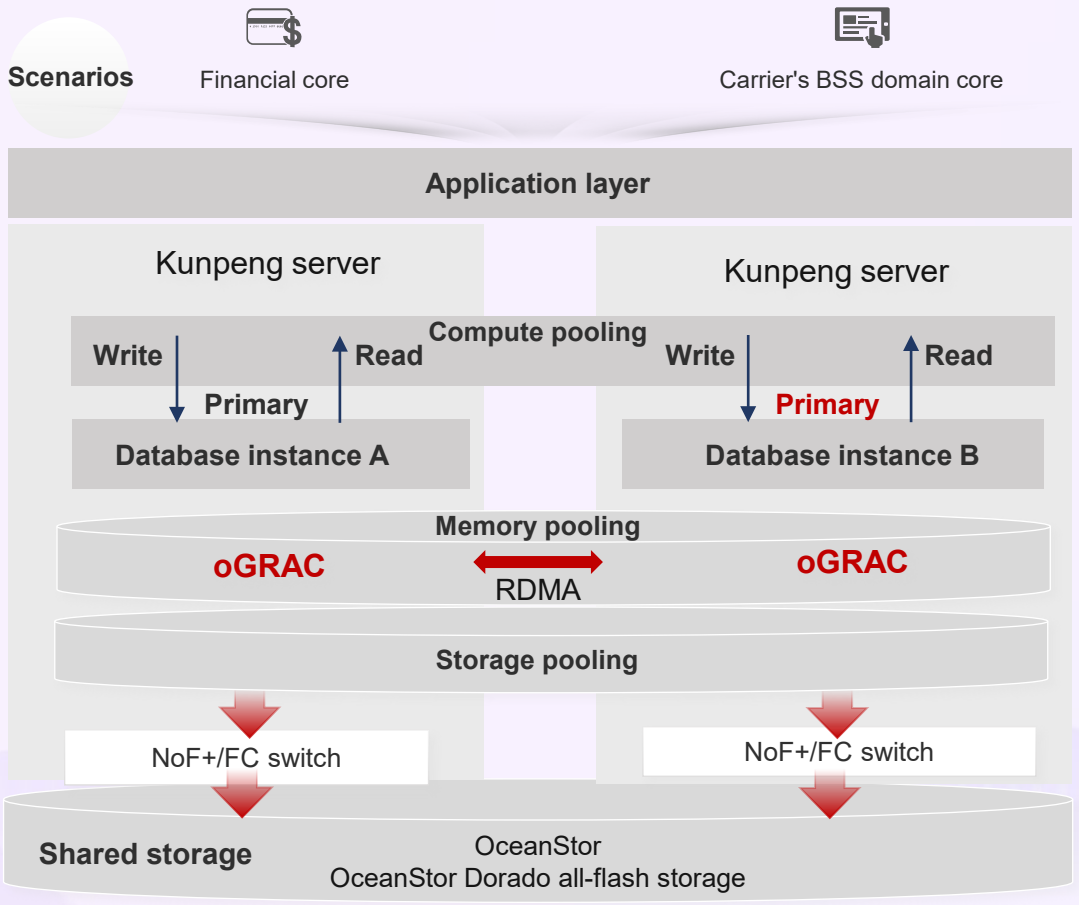
04

Multi-primary clusters: rapid node failover, ensuring E2E high availability of clusters

openGauss

Kunpeng + openGauss oGRAC Multi-Write Solution Doubling Performance

oGRAC multi-write database solution



Benefits

1. **Leading performance**

   3.5 million tpmC
   Dual-node TPCC

   5% ↑
   vs. peers

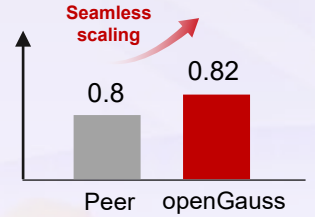
2. **Second-level failover for financial-grade reliability**

   RPO = 0, RTO < 10s

3. **50%+ savings in compute and storage resources**

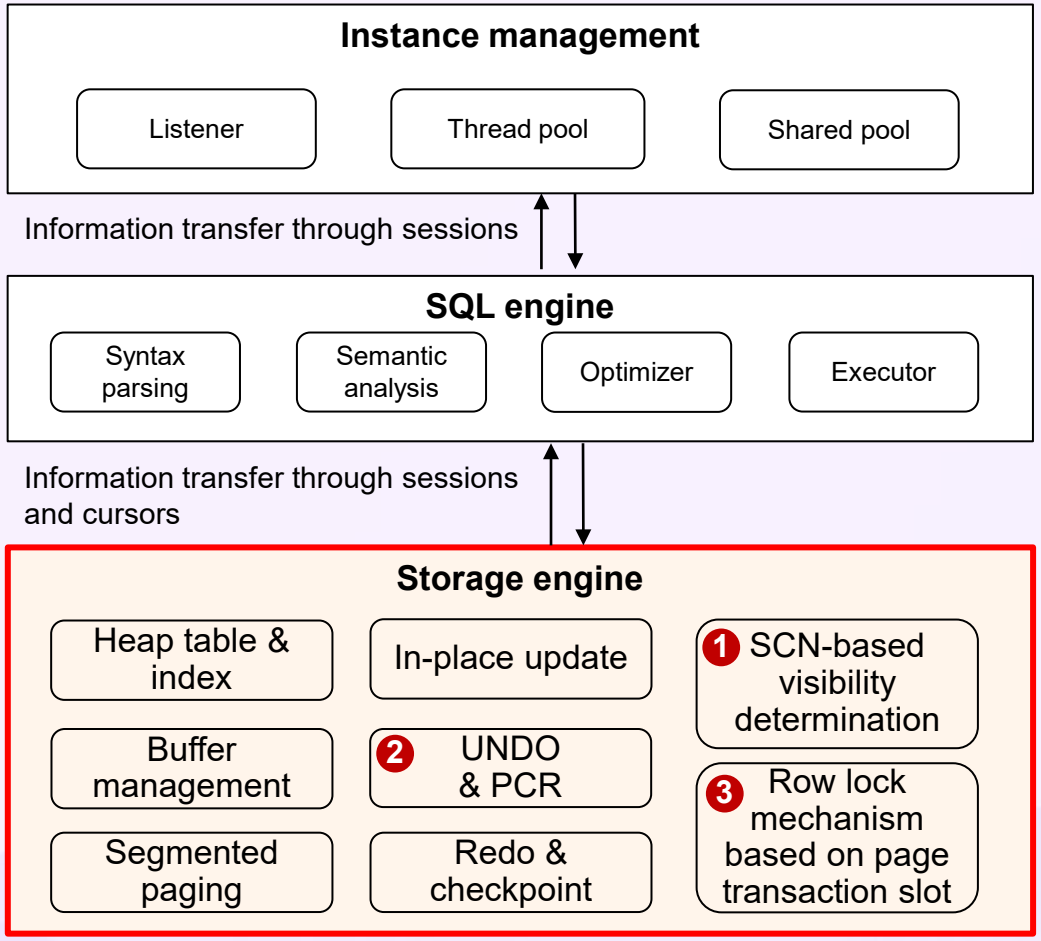
   Based on a three-tier pooling architecture, compute, memory, and storage resources are shared across nodes, reducing resource consumption by at least 50% for the same workload.

4. **High expansion ratio**



openGauss

Storage Engine: Naturally Extensible to a Multi-Write Architecture



Unique identifier for visibility determination

Page-level multi-version concurrency control

Row lock mechanism

1 Transaction visibility determination using timestamp-based auto-incrementing SCN

- SCN serves as the global logical timestamp to control the transaction commit sequence.
- SCN is used as the only resource to determine transaction visibility, eliminating the need to use transaction IDs as a cluster's global resource.

2 In-place update of block-level MVCC & UNDO-based page consistent read (PCR)

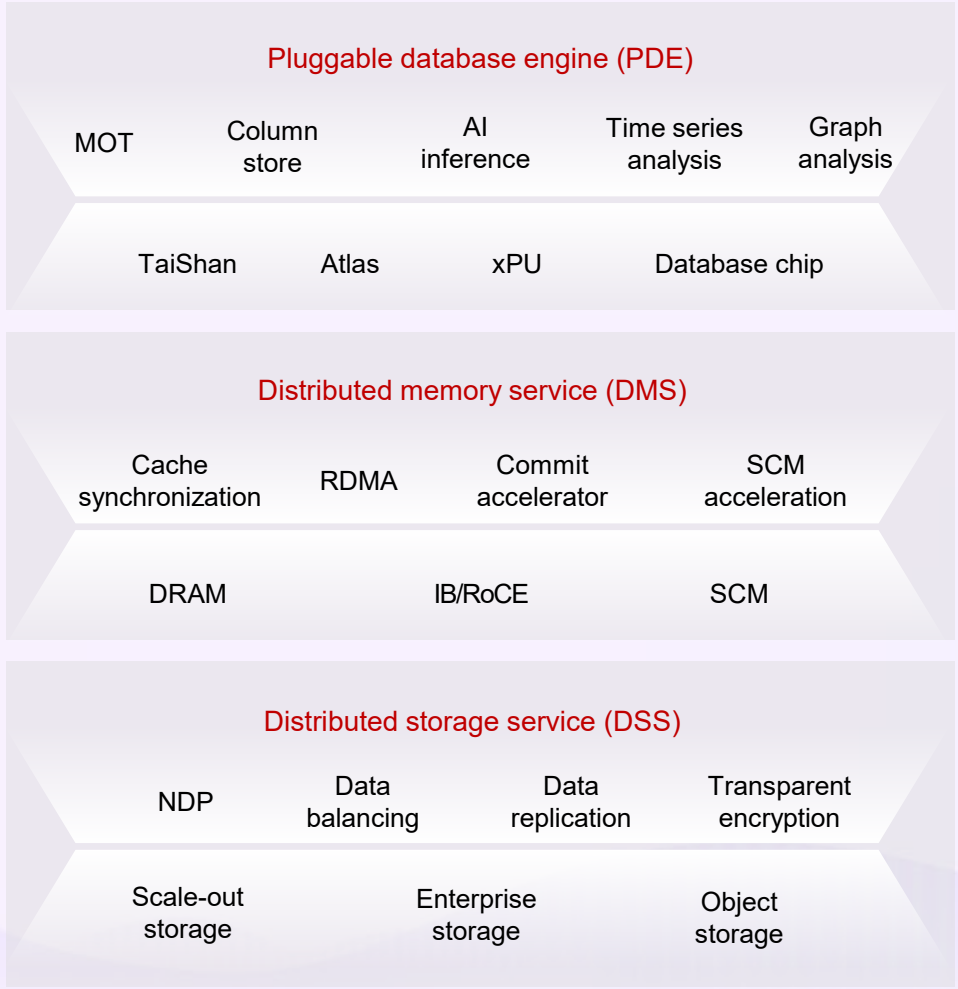
- Historical version records of data are stored in the UNDO space, and data pages are updated in-place, so the data page size does not grow.
- A consistent page of a given snapshot is constructed based on UNDO records. The current version page only needs to be accessed once, reducing read/write conflicts.
- CR pages, not data rows, are transferred across nodes, greatly reducing interactions.

3 Row lock mechanism based on page transaction slot

- The page supports a physical transaction slot. When a transaction commits, the SCN and transaction status are backfilled into the page transaction slot. Transaction information is transferred as the page is transmitted across nodes.
- Row locks and transaction locks are not required, improving the parallel transaction execution efficiency.
- Transaction status is centrally managed in a transaction table.

Directory

DataPod: Transparently Scalable Resource Pooling Architecture with Hardware-Software Collaboration for Full-Stack Optimization



Compute pooling

Microkernel for diversified computing power

- Database chip for TP/AP acceleration to improve energy efficiency
- AI training and inference for Atlas and others
- SPQ multi-node parallelism for 2.6x+ TPC-H/TPC-DS performance
- Transparent forwarding of write operations

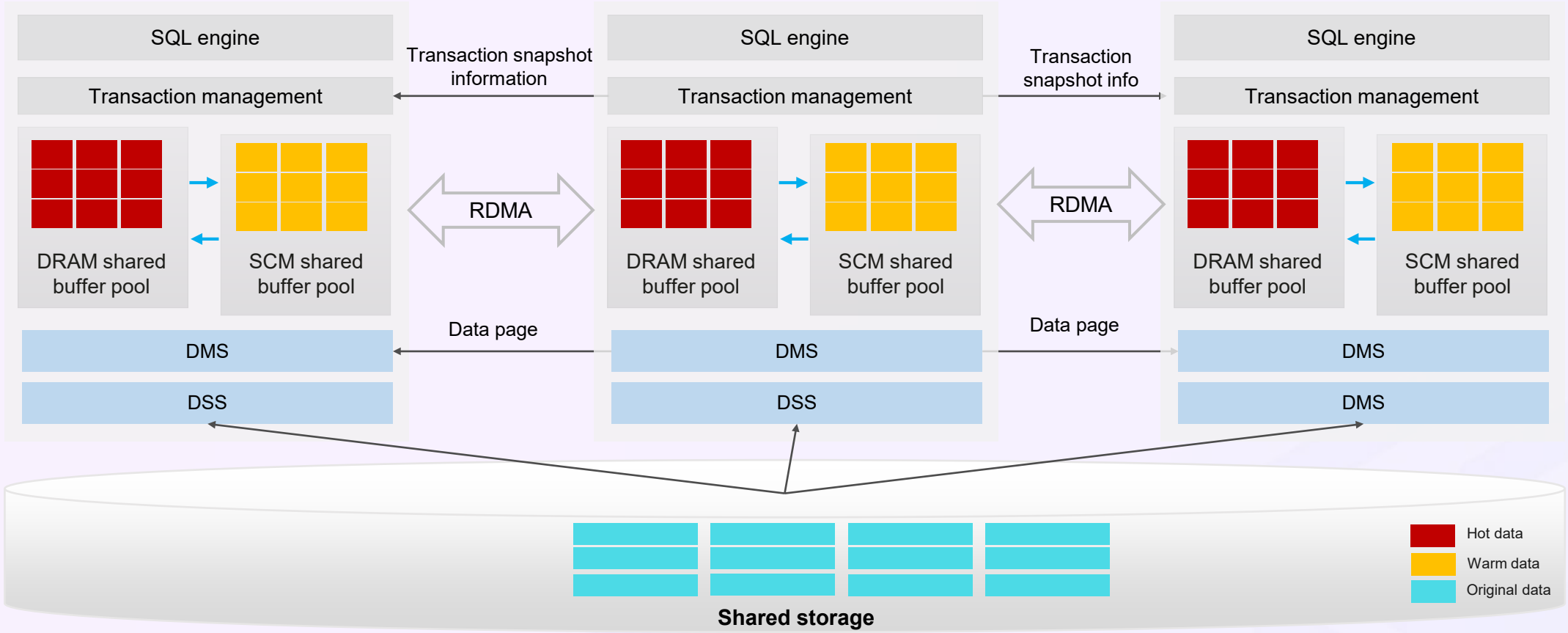
Memory pooling

- High-speed network interconnection
- Database transaction synchronization and multi-node memory pooling (8 nodes)
- Persistent-memory-based SCM, data cache acceleration, and large-capacity column store
- 1.5x+ scale-out efficiency, 10x complex query performance

Storage pooling

- Shared storage operator offloading and storage replication-based intra-city HA
- Near data computing, supporting multiple types of storage
- On-demand Xlog playback in a cluster, with RTO < 10s under typical workloads
- Dual clusters based on OceanStor Dorado synchronous replication, with RTO < 30s under typical workloads
- Virtualized deployment and cloud integration

Memory Pooling: Multi-Node Real-Time Data Consistency, RDMA-Based Network Acceleration, and SCM-Based Multi-Level Caching



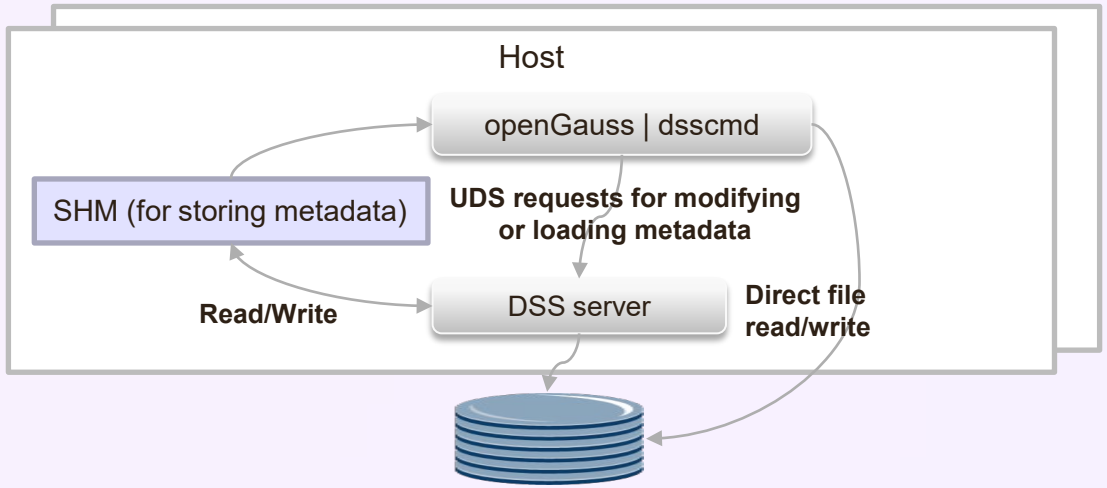
The cluster supports:

- **1 primary node** for read and write and **7 standby nodes** for **scale-out read**, meeting real-world **performance requirements of typical workloads**
- Multi-node real-time data consistency for **transparent scaling** of data consistency-sensitive applications from a single node **to multiple nodes**
- **SCM-based multi-level caching** for 30% performance boost at the same memory cost
- Seamless service failover to a standby node upon a primary node failure, with RTO < 10s and RPO = 0

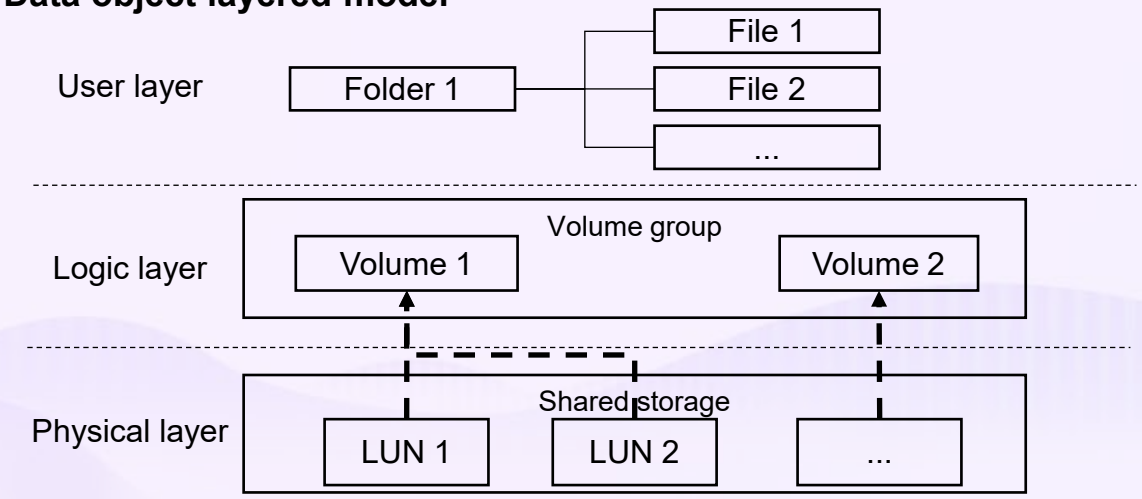
openGauss

Storage Pooling: DSS Manages Raw Devices to Build a Userspace Filesystem for Multi-Node Shared Storage

Multi-node file system



Data object layered model



Pain points

Cache overheads due to data access through the file system, adversely impacting the disk throughput

Key tech

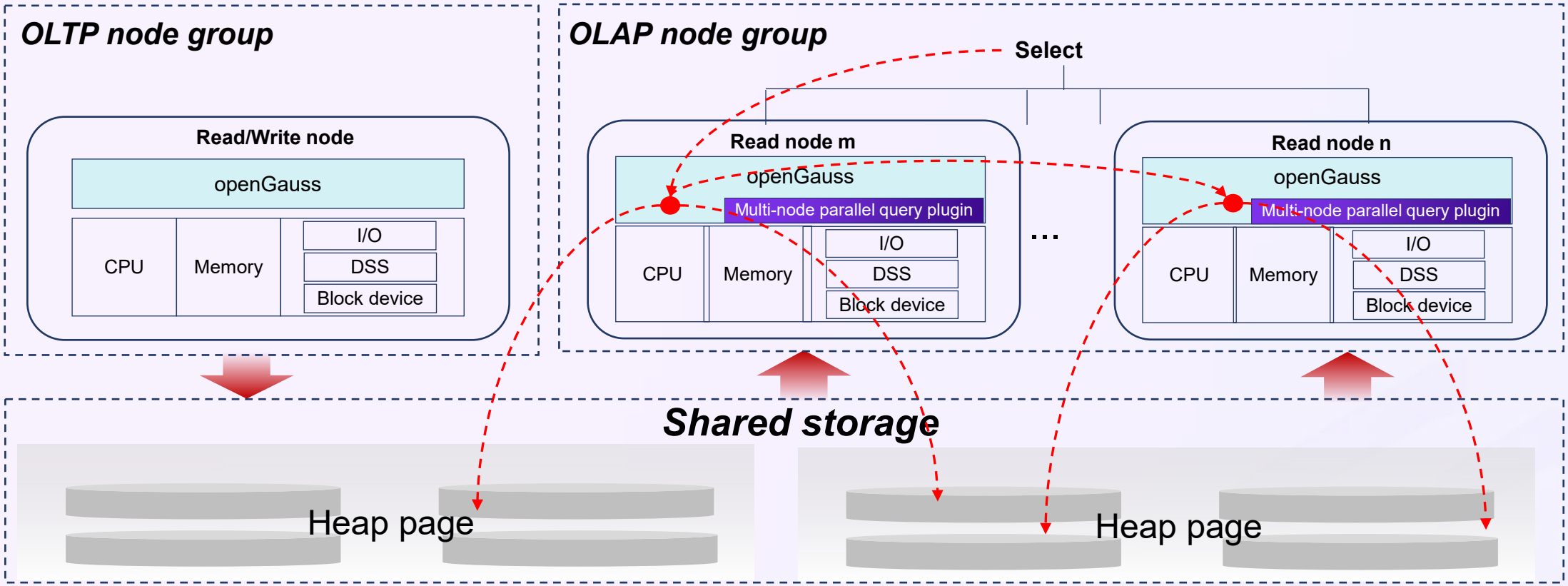
- Direct management of raw devices by volume groups, bypassing the file system layer (data directly transmitted from disks to applications, reducing cache overheads of the file system and improving the disk throughput through asynchronous I/O)
- NoF+ protocol for storage access
- File metadata persistence mechanism based on redo logs, ensuring a highly reliable file system
- APIs to create, delete, and access volumes, volume groups, directories, and files, facilitating efficient development
- Tools to manage the virtual file system

Benefits

- 30% performance gain with NoF+ protocol vs. FC
- 8 volume groups per host, up to 8 PB per volume group, up to 8 TB per file
- Black box logging

openGauss

Compute Pooling: Multi-Node Parallel Query and Resource Collaboration to Improve Complex AP Query



Pain points

Waste of computing resources (Upon large-scale data queries on a read node, only this node's computing resource is used. Computing resources of other nodes remain idle.)

Key tech

- Optimizer generation of multi-node execution plans
- Plan distribution to nodes for synchronous execution and data aggregation by the querying node to implement parallel query on all read nodes

Benefits

- 2x+ TPC-H/TPC-DS performance in complex AP queries with multi-node parallel plan generation and reduced inter-node data transmissions
- 2x index creation performance with multi-node parallel index creation

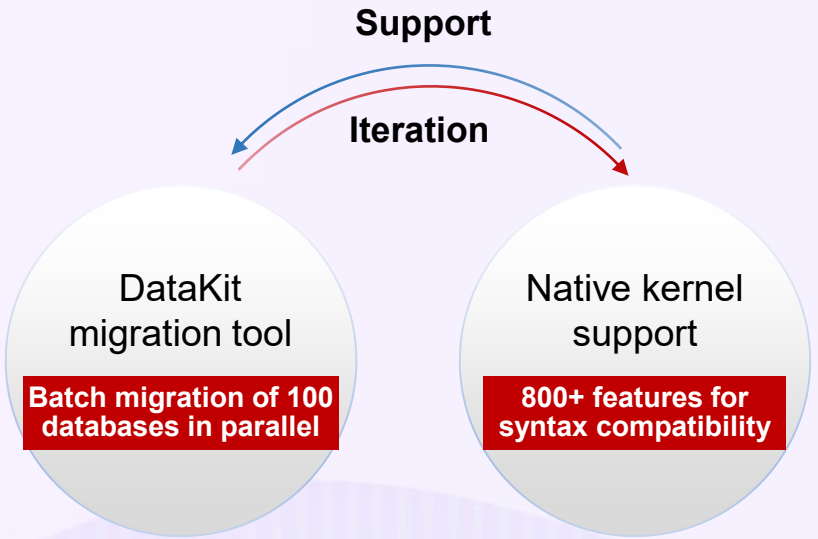
Directory

Database Migration: Comprehensive Upgrade of Syntax Compatibility for Fast Data Migration and Verification

Migration evaluation	Full migration	Incremental migration	Workload replay	Data verification	Parallel running
Automatic SQL extraction and evaluation	1 TB of data migrated within an hour	30,000 TPS on a 2-socket server	Early detection of slow SQL statements	Real-time detection of inconsistencies	Quick escape

openGauss migration tool set

- MySQL migration
- PostgreSQL migration
- New SQL Server migration
- Elasticsearch/Milvus migration New
- Migration task orchestration
- Data recording and replay
- Full migration
- Incremental migration
- Reverse migration
- Data verification

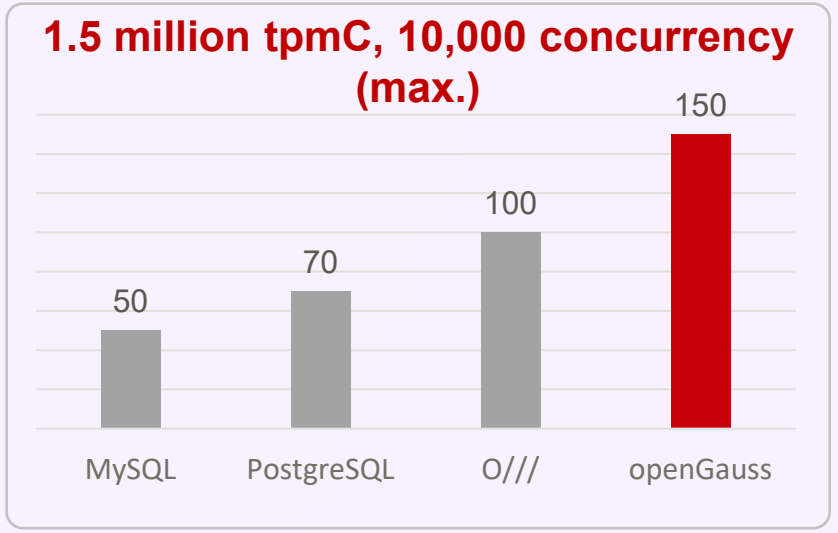


openGauss plug-in architecture

- MySQL compatibility evaluation tool
- openGauss connection driver
- MySQL syntax compatibility plug-in
- openGauss SQL engine
- SQL engine plug-in architecture
- Storage engine of openGauss

Directory

Single-Node: Row-Store Engine-based TPC-C



* 1.5 million tpmC performance: test result of the openGauss community edition on the 2-socket Kunpeng 920 CPU-powered server

* 10,000 concurrency: the highest concurrency supported by the 2-socket Kunpeng 920 CPU-powered server

* Performance on a par with O/// in the x86 environment; 50% higher performance than that of O/// in the Kunpeng environment.

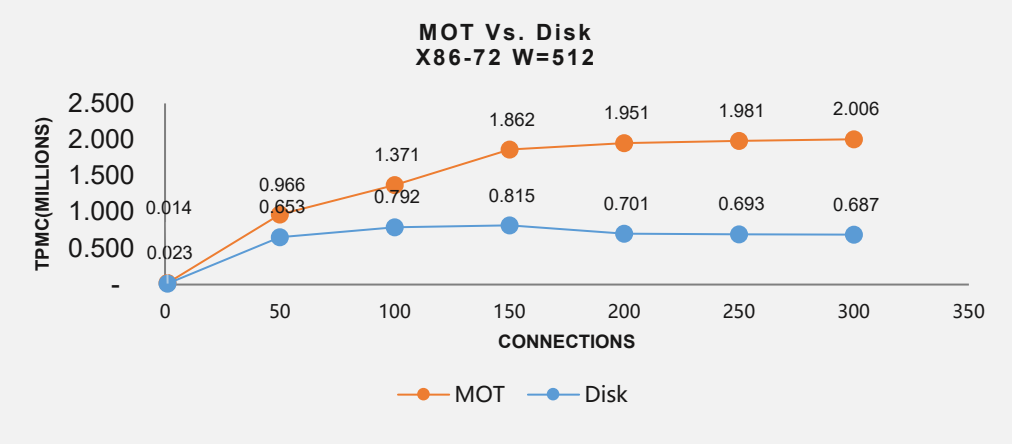
openGauss performance improves with higher concurrency, remaining comparable to Database A at 80 or fewer concurrent requests.

1. **Robust at low concurrency (80/160/240):** Both openGauss and Database A outperform MySQL and PostgreSQL.
2. **High concurrency advantage:**
   - At 160 concurrent requests: openGauss's performance is 1.1x that of Database A.
   - At 2,000 concurrent requests: openGauss's performance is 1.8x that of Database A.
3. **Stable under extreme load:** Beyond 2,000 concurrent requests, Database A's performance degrades significantly, whereas openGauss remains stable.

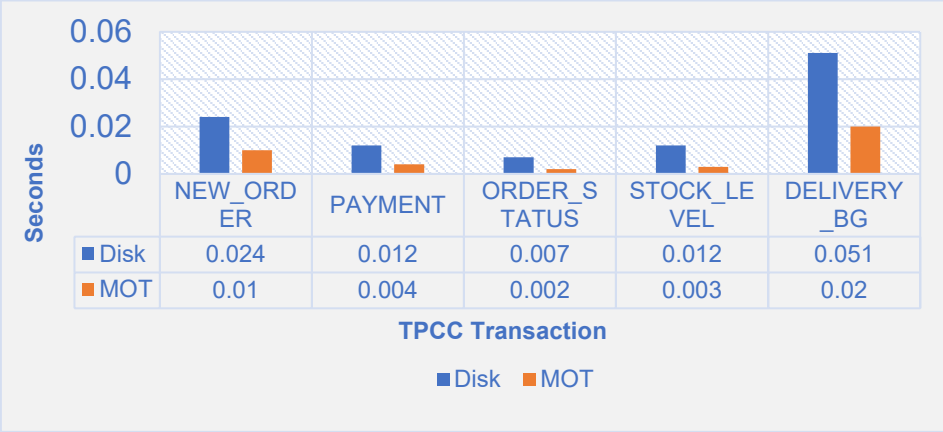
Test Category	Test Scenario	openGauss	Database A	MySQL	PostgreSQL	Database B
Standard TPC-C transaction test	80 concurrent requests	851267	885696	353309	385605	533031
	160 concurrent requests	1374697	1208984	151162	494512	526943
	240 concurrent requests	1431903	1362022	140133	513978	539086
	2,000 concurrent requests	1420305	796093	62858	363210	339645

Single-Node: MOT Engine-based TPC-C

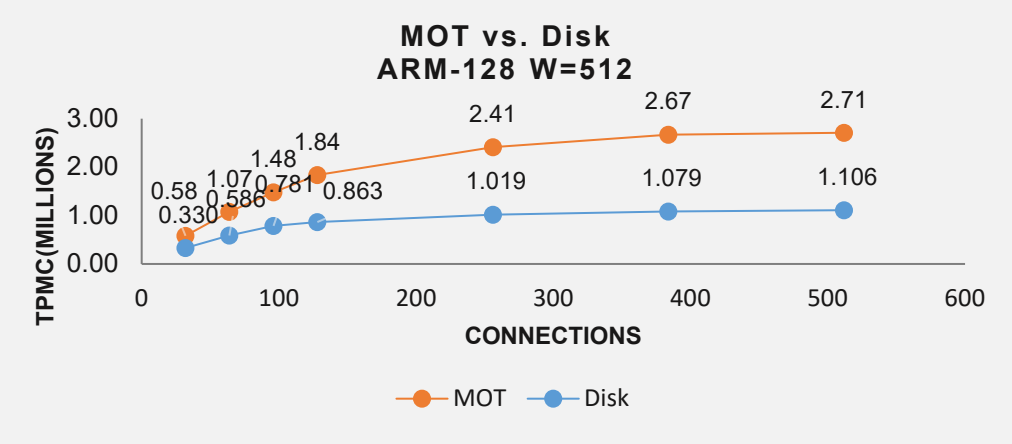
TPCC Throughput on x86-72 vCores



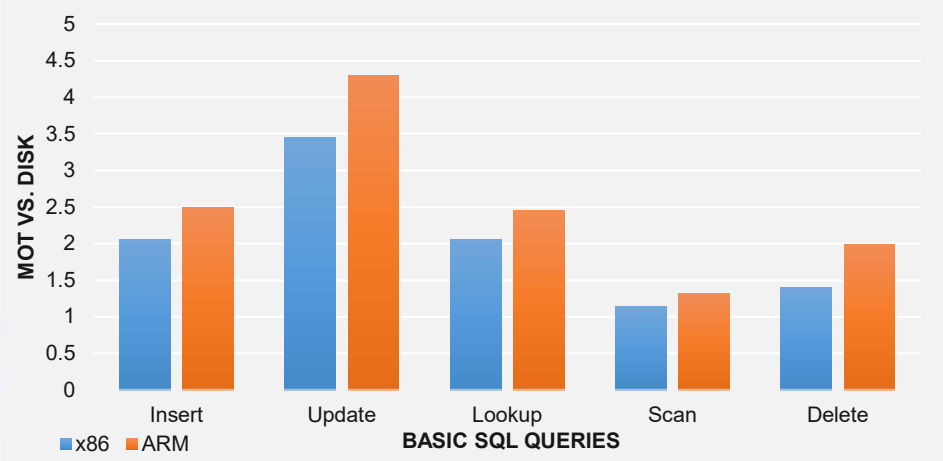
TPCC Latency



TPCC Throughput on Arm-128 Cores



Basic SQL Speed Up



openGauss

openGauss

THANKS