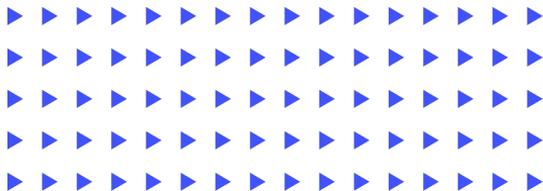


# Apache ShardingSphere 5.0.0 全新 Database Plus 架构演进

端正强  
SphereEx 高级中间件工程师



# 个人介绍



**端正强**

SphereEx 高级中间件工程师  
Apache ShardingSphere Committer

- ◆ 2018 年开始接触 Apache ShardingSphere，曾主导公司内部海量数据的分库分表，在数据库、中间件的开发等有着丰富的实践经验；
- ◆ 热爱开源，乐于分享，目前专注于 Apache ShardingSphere 内核模块开发；

# 目录

---

1. ShardingSphere 简介
2. ShardingSphere Database Plus 架构解读
3. ShardingSphere Database Plus 解决方案
4. ShardingSphere 未来规划

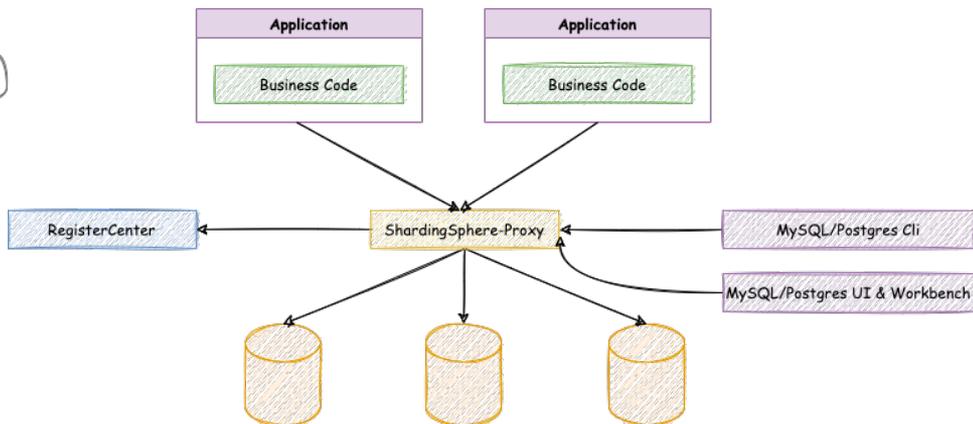
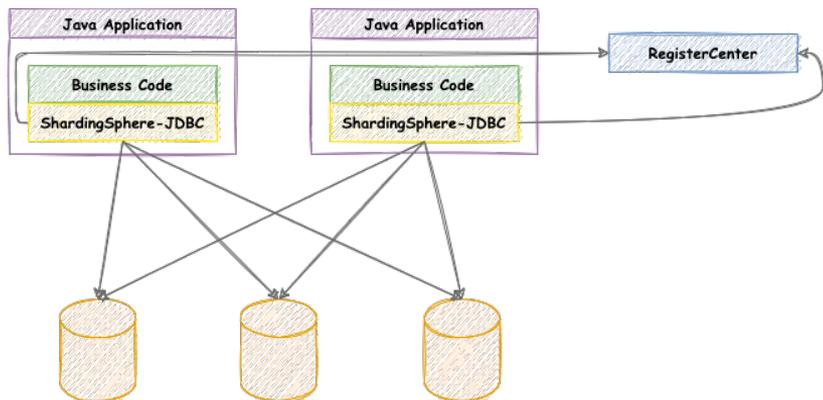
# ShardingSphere 简介

ShardingSphere 是一款**开源的数据服务能力增强引擎**，提供数据分片、分布式事务、数据安全等能力。

ShardingSphere 遵循 Database Plus 理念，旨在构建异构数据库上层的服务标准和生态。

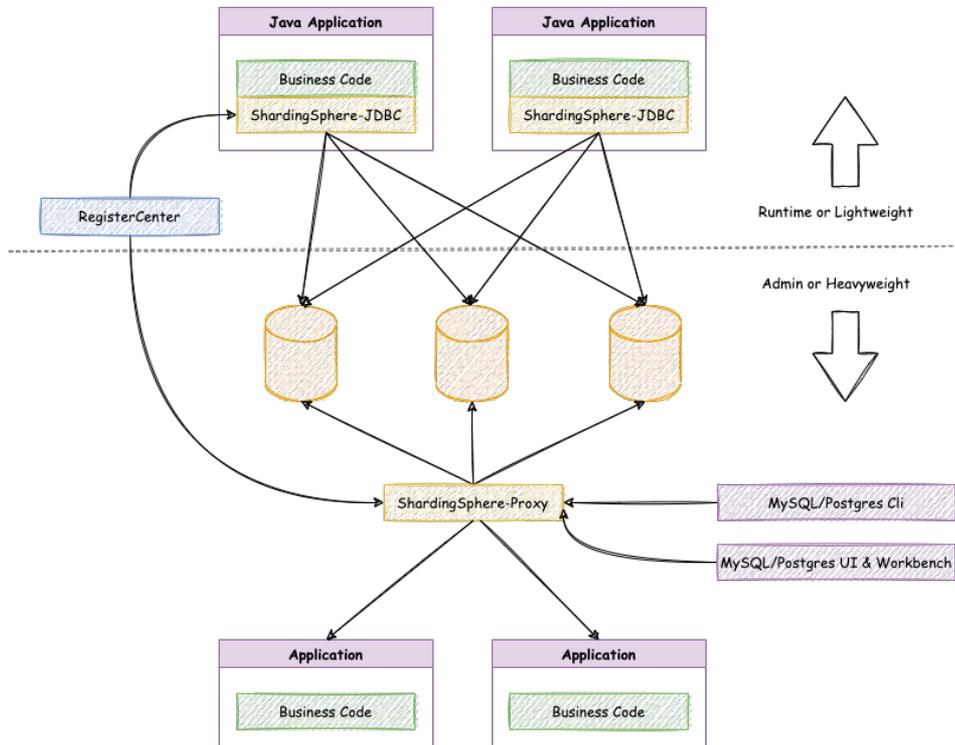


# ShardingSphere 接入端



- ◆ ShardingSphere 由 JDBC、Proxy 和 Sidecar（规划中）3 个接入端组成，既支持独立部署，又可以混合部署；
- ◆ JDBC、Proxy 和 Sidecar 3 个接入端均提供标准化的增量功能，可适用于 Java 同构、异构语言以及云原生等各种应用场景；

# ShardingSphere 接入端混合部署



- ◆ ShardingSphere-JDBC 采用无中心化架构，与应用程序共享资源，适用于 Java 开发的**高性能的轻量级 OLTP 应用**；
- ◆ ShardingSphere-Proxy 提供静态入口以及异构语言的支持，独立于应用程序部署，**适用于 OLAP 应用**以及对分片数据库进行管理和运维的场景。

# ShardingSphere 社区

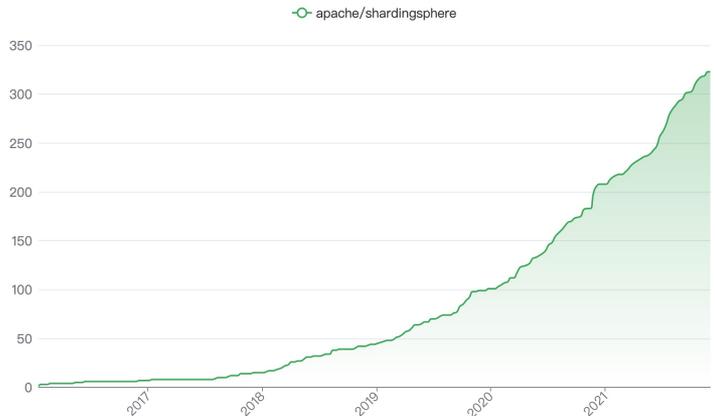
- ◆ 官网: <https://shardingsphere.apache.org>
  - ◆ 中文论坛: <https://community.sphere-ex.com>
  - ◆ GitHub: <https://github.com/apache/shardingsphere>
  - ◆ GitHub stars: 15k+
  - ◆ 贡献者数量: 300+
  - ◆ 模块数量: 190+
  - ◆ 采用公司: 170+
- ◆ Apache 顶级开源项目
  - ◆ CNCF 全景图项目
  - ◆ 鲲鹏技术认证项目
  - ◆ 中国信息通讯研究院可信开源社区认证项目
  - ◆ 中国信息通讯研究院可信开源项目认证
  - ◆ 中国开源云联盟, 优秀开源项目
  - ◆ 2020 年度国人主导的最活跃的 Apache 项目



# ShardingSphere 社区

贡献者增长时间线

Contributor Over Time



- ◆ 社区优于代码
- ◆ 尊重和信任合作伙伴
- ◆ 保持友好和开放

November 4, 2021 – December 4, 2021

Period: 1 month

## Overview

257 Active Pull Requests

255 Active Issues

250

Merged Pull Requests

7

Open Pull Requests

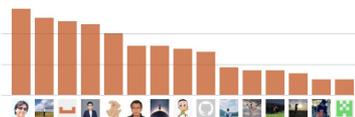
167

Closed Issues

88

New Issues

Excluding merges, **39 authors** have pushed **250 commits** to master and **250 commits** to all branches. On master, **2,275 files** have changed and there have been **38,759 additions** and **14,122 deletions**.



1 Release published by 1 person

5.0.0

published 25 days ago

- ◆ 测试自动化
- ◆ 创建自动渠道
- ◆ 公开和远程的工作方式

# ShardingSphere 社区



DZone Database Zone

Over 2 million developers have joined DZone. [Log In / Join](#)

REFCARDZ RESEARCH WEBINARS [Agile](#) [AI](#) [Big Data](#) [Cloud](#) [Database](#) [DevOps](#) [Integration](#) [IoT](#) [Java](#) [Microservices](#) [Open Source](#) [Performance](#) [Security](#) [Web Dev](#)

DZone > Database Zone > Apache ShardingSphere 5.0.0 New Features: Middleware to Ecosystem Evolution

## Apache ShardingSphere 5.0.0 New Features: Middleware to Ecosystem Evolution

ShardingSphere recreates the distributed pluggable system, connecting user implementation scenarios, and contributing valuable solutions to the database industry.



by Veronica Xu · Nov. 16, 21 · Database Zone · Presentation

[Like \(2\)](#) [Comment \(0\)](#) [Save](#) [Tweet](#) 3,91K Views

Join the DZone community and get the full member experience. [JOIN FOR FREE](#)

Google Summer of Code

DASHBOARD **PROJECTS** | [User](#) | [Menu](#)

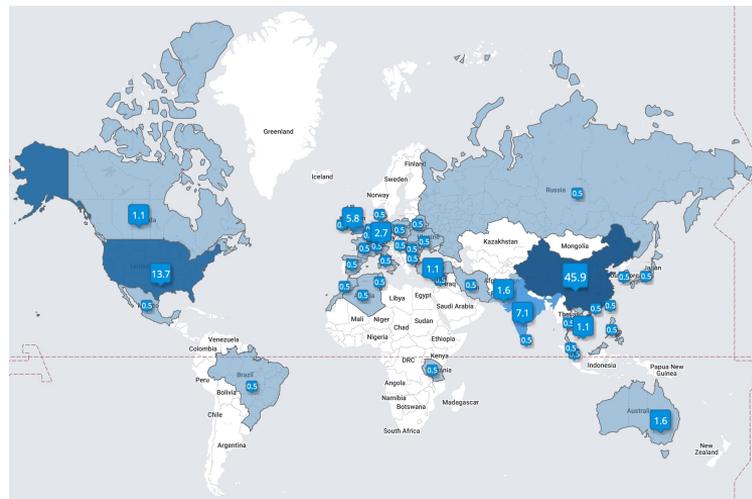
## Projects

ALL **MY PROJECTS**

Search for a project

ACTIVE

Display Name	Title ↑	Organization	Active	Last Update
Liangda Wang	<a href="#">Apache ShardingSphere - Proofread the Oracle DDL/TCL SQL Definitions</a>	<a href="#">The Apache Software Foundation</a>	Yes	April 10, 2021
Thanoshan M.V.	<a href="#">Proofread Oracle DDL SQL Definitions for the ShardingSphere Parser</a>	<a href="#">The Apache Software Foundation</a>	Yes	April 5, 2021

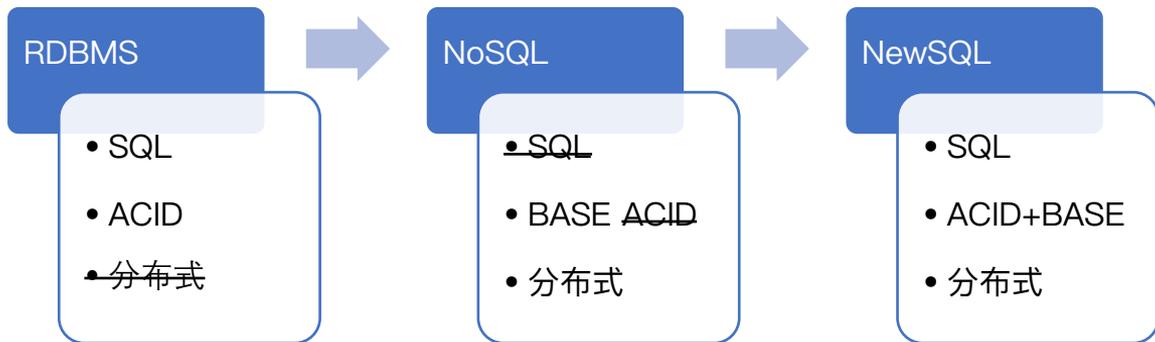


# 目录

---

1. ShardingSphere 简介
2. ShardingSphere Database Plus 架构解读
3. ShardingSphere Database Plus 解决方案
4. ShardingSphere 未来规划

# 数据库行业发展现状



- ◆ 单机 -> 分布式：RDBMS -> NoSQL -> NewSQL
- ◆ 单一化 -> 多样化：单一数据库满足各种场景 -> 根据多样化场景选择不同类型数据库

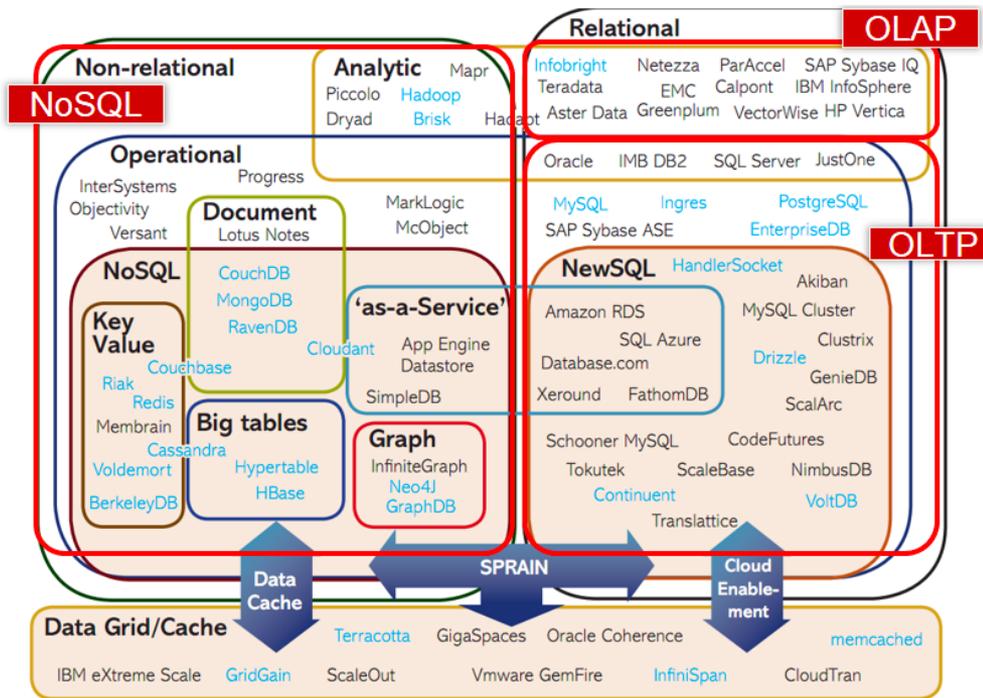


- ◆ New Architectures
- ◆ **Transparent Sharding Middleware**
- ◆ Database-as-a-Service

Andrew Pavlo—What's Really New with NewSQL?

<https://db.cs.cmu.edu/papers/2016/pavlo-newsq-sigmodrec2016.pdf>

# 数据库行业发展现状



- ◆ 数据库系统百花齐放，碎片化趋势明显；
- ◆ 异构数据库访问协议、SQL 方言、存储接口和结构均不同；
- ◆ 异构数据库间无协作能力；
- ◆ 异构数据库间的共性需求无法满足；

# Database Plus 架构解读

Database Plus 是分布式数据库系统的一种设计理念。用于在关系型数据库，如 MySQL，PostgreSQL，Oracle，SQLServer，OpenGauss 之上为用户 **提供数据分片、数据加解密等增强能力**。

通过 **在碎片化数据库之上搭建使用和交互的标准层和生态层**，使得所有应用和数据库之间的交互，面向 Database Plus 构建的标准层，从而屏蔽底层不同数据库对上层业务的影响和差异。

# Database Plus 架构解读

产品定位:

- ◆ 构建异构数据库的上层标准和生态;
- ◆ 提供精准化和差异化的能力;

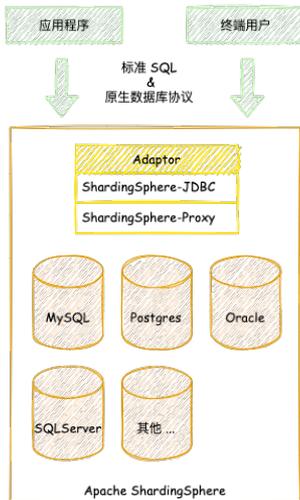
核心能力: 连接、增量、可插拔



# Database Plus 架构解读

## Database Plus

构建多模数据库上层的标准和生态



## 连接

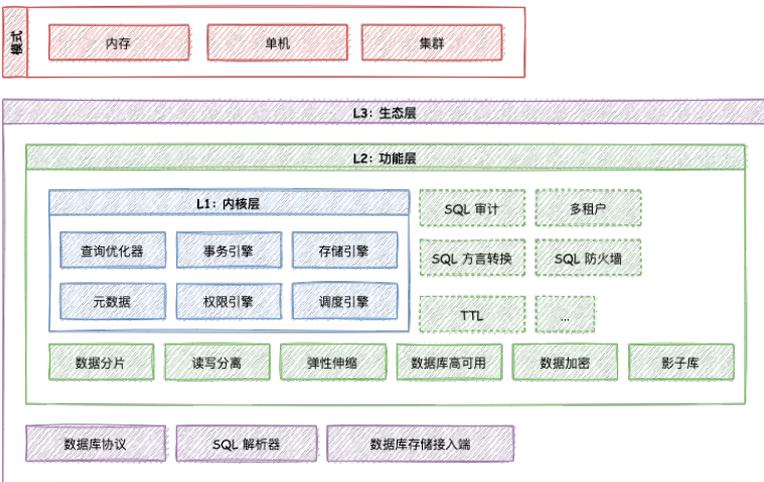
连接数据和应用, 关注多模数据库之间的合作

## 增量

通过数据库入口流量的抓取提供透明化的增量功能

## 可插拔

微内核完全面向可插拔的三层架构体系设计



- ◆ L1 内核层: 面向数据库内核, 包括数据库事务引擎, 查询优化器等;
- ◆ L2 功能层: ShardingSphere 最核心所在, 可定制化开发平台。具有高定制化、高度内聚、灵活扩展等特点;
- ◆ L3 生态层: 通过三个接口分别实现数据库协议、SQL 方言和数据库存储对接, 用于打造异构数据网关;

# Database Plus 架构解读



## 连接

连接是 ShardingSphere 的基础能力，可以有效**简化数据和应用之间的连接**。连接的设计要点在于**强大的数据库的兼容性**，在应用和数据之间搭建了一层与具体数据库实现无关的桥梁，为增量能力提供了基础。



## 增强

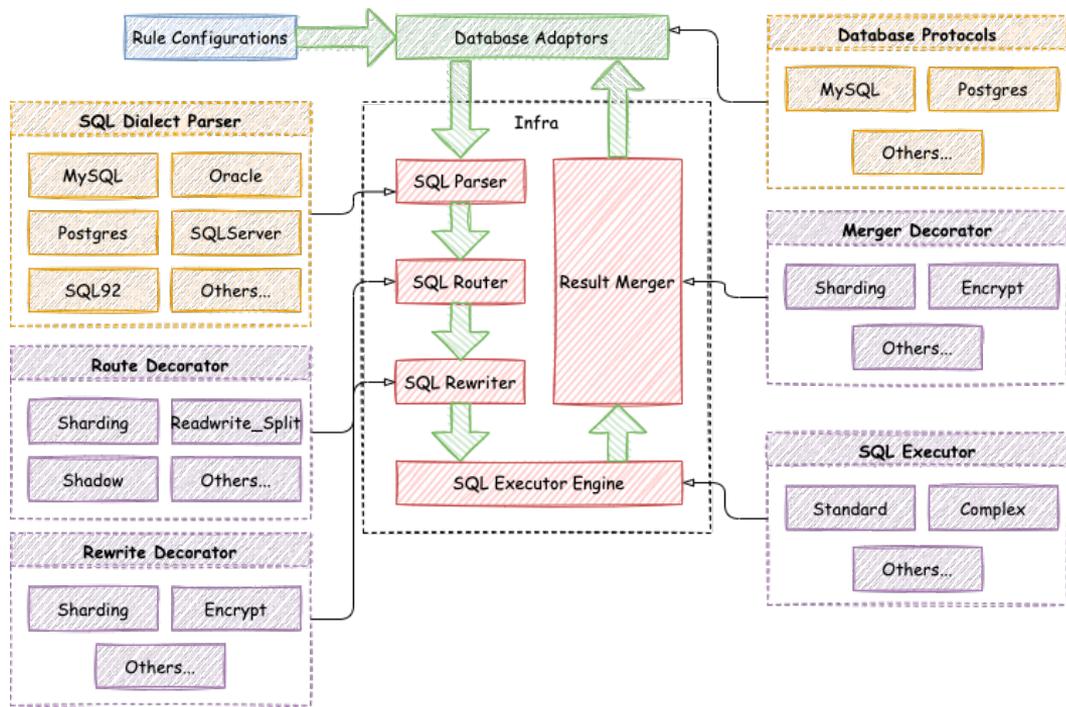
增强是 ShardingSphere 的主要能力，在拦截访问数据库流量的前提下，透明化的提供增量功能。增强包含了**流量的重定向**（数据分片、读写分离、影子库）、**流量变形**（数据加密）、**流量鉴权**（SQL 审计、权限）、**流量治理**（熔断、限流）以及**流量分析**（可观察性、服务质量分析）等。



## 可插拔

可插拔是 ShardingSphere 的设计理念，架构内核是**完全面向顶层接口设计的**，**内核模块完全不感知具体功能的存在**。它为分库分表、读写分离等每一个功能插件赋予单独部署和协同配合的能力。

# Database Plus 架构解读



- ◆ ShardingSphere 可插拔架构提供了数十个基于 SPI 的扩展点，开发者可以十分方便的对功能进行定制化扩展；
- ◆ 按照扩展点是基于技术还是基于功能实现，可以将扩展点划分为功能扩展点和技术扩展点。
- ◆ 基于扩展点，ShardingSphere 默认实现了数据分片、读写分离、数据加密、影子库压测、高可用等功能；

- ◆ 数据分片：实现了元数据加载、SQL 路由、SQL 改写和结果归并的全部功能扩展点，在数据分片功能内部，又提供了分片算法、分布式 ID 等扩展点；
- ◆ 读写分离：实现了 SQL 路由的功能扩展点，功能内部提供了负载均衡算法扩展点；
- ◆ 数据加密：实现了元数据加载、SQL 改写和结果归并的扩展点，内部提供了加解密算法扩展点；
- ◆ 影子库压测：实现了 SQL 路由的扩展点，在影子库压测功能内部，提供了影子算法扩展点；
- ◆ 高可用：实现了 SQL 路由的扩展点；

# 目录

---

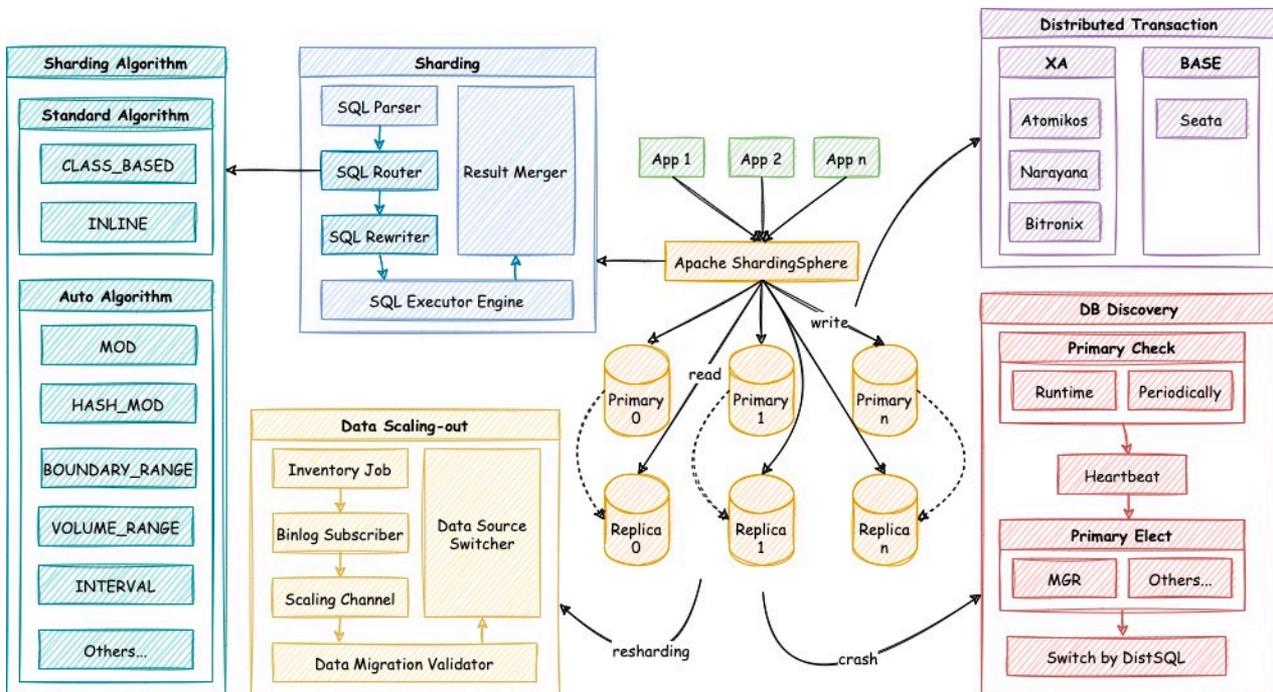
1. ShardingSphere 简介
2. ShardingSphere Database Plus 架构解读
3. ShardingSphere Database Plus 解决方案
4. ShardingSphere 未来规划

# Database Plus 解决方案

基于 Database Plus 架构以及可插拔内核，ShardingSphere 提供了丰富的功能，如数据分片、读写分离及数据加密等。基于这些丰富的功能，ShardingSphere 在产品层面也提供了**分布式数据库、数据安全、数据库网关和全链路压测** 4 套完善的解决方案。

分布式数据库	数据安全	数据库网关	全链路压测
数据分片	数据加密	异构数据库支持	影子库
读写分离	行级权限 (TODO)	SQL 方言转换 (TODO)	可观测性
分布式事务	SQL 审计 (TODO)		
弹性伸缩	SQL 防火墙 (TODO)		
高可用			

# Database Plus 解决方案-分布式数据库



# Database Plus 解决方案-分布式数据库

```
mysql> CREATE DATABASE sharding_db;
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> USE sharding_db;
Database changed
```

```
mysql> ADD RESOURCE ds_0 (
  -> HOST=127.0.0.1,
  -> PORT=3306,
  -> DB=demo_ds_0,
  -> USER=root,
  -> PASSWORD=123456
  -> );
Query OK, 0 rows affected (0.96 sec)
```

创建逻辑数据库并添加资源

```
mysql> ADD RESOURCE ds_1 (
  -> HOST=127.0.0.1,
  -> PORT=3306,
  -> DB=demo_ds_1,
  -> USER=root,
  -> PASSWORD=123456
  -> );
Query OK, 0 rows affected (0.10 sec)
```

```
mysql>
```

```
mysql> CREATE SHARDING TABLE RULE t_order(
  -> RESOURCES(ds_0,ds_1),
  -> SHARDING_COLUMN=order_id,
  -> TYPE(NAME=hash_mod,PROPERTIES("sharding-count"=4)),
  -> GENERATED_KEY(COLUMN=order_id,TYPE(NAME=snowflake,PROPERTIES("worker-id"=123)))
  -> );
Query OK, 0 rows affected (1.38 sec)
```

创建分片规则和分片表

```
mysql> CREATE TABLE `t_order` (
  -> `order_id` int NOT NULL,
  -> `user_id` int NOT NULL,
  -> `status` varchar(45) DEFAULT NULL,
  -> PRIMARY KEY (`order_id`)
  -> ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
mysql> SHOW TABLES LIKE 't_order%';
```

```
+-----+
| Tables_in_demo_ds_0 (t_order%) |
+-----+
t_order_0
t_order_2
```

分片表对应的真实表

# Database Plus 解决方案-分布式数据库

```
mysql> BEGIN;
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO t_order(order_id, user_id, status) VALUES(1, 1, 'OK'), (2, 1, 'DOWN');
Query OK, 2 rows affected (0.40 sec)

mysql> SELECT * FROM t_order;
+-----+-----+-----+
| order_id | user_id | status |
+-----+-----+-----+
| 2 | 1 | DOWN |
| 1 | 1 | OK |
+-----+-----+-----+
2 rows in set (0.11 sec)

mysql> ROLLBACK;
Query OK, 0 rows affected (0.02 sec)

mysql> SELECT * FROM t_order;
Empty set (0.03 sec)

mysql> INSERT INTO t_order(order_id, user_id, status) VALUES(1, 1, 'OK'), (2, 1, 'DOWN');
Query OK, 2 rows affected (0.01 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM t_order;
+-----+-----+-----+
| order_id | user_id | status |
+-----+-----+-----+
| 2 | 1 | DOWN |
| 1 | 1 | OK |
+-----+-----+-----+
2 rows in set (0.01 sec)
```

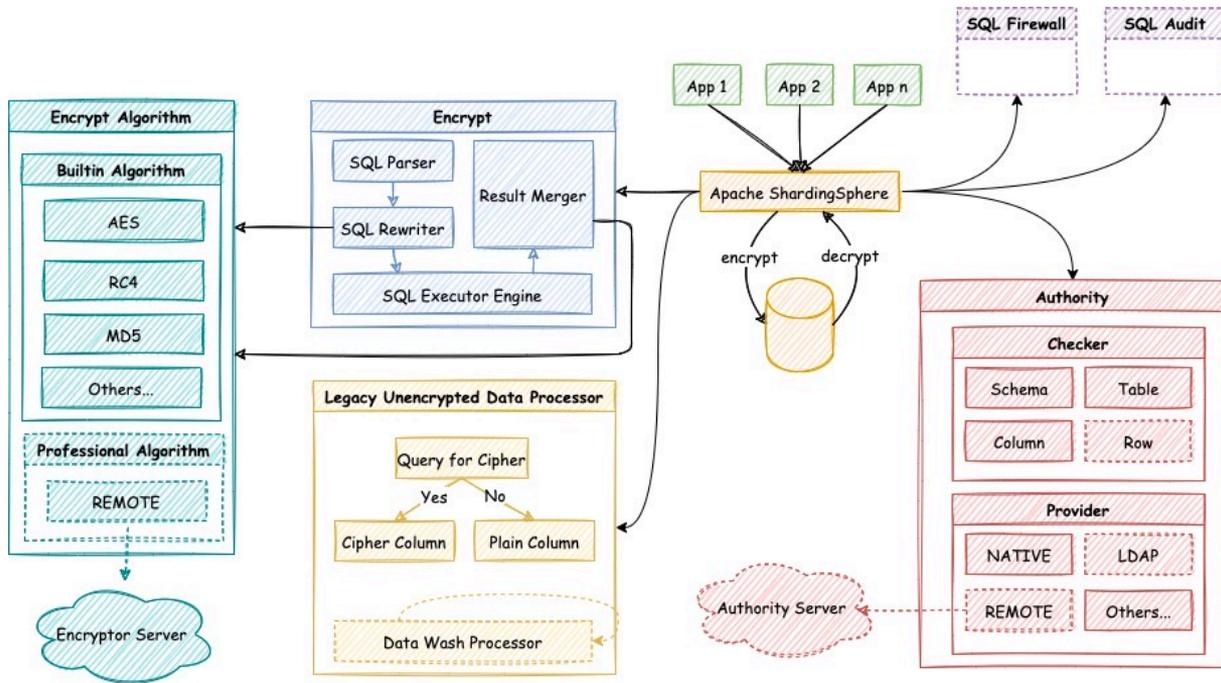
• 支持 XA 和 Base 事务

```
mysql> ALTER SHARDING TABLE RULE t_order (
-> RESOURCES(ds_0, ds_1, ds_2, ds_3),
-> SHARDING_COLUMN=order_id,
-> TYPE(NAME=hash_mod,PROPERTIES("sharding-count"=8)),
-> GENERATED_KEY(COLUMN=order_id,TYPE(NAME=snowflake,PROPERTIES("worker-id"=123)))
-> );
Query OK, 0 rows affected (0.62 sec)

mysql> SHOW SCALING LIST;
+-----+-----+-----+-----+-----+-----+
| id | tables | sharding_total_count | active | create_time | stop_time |
+-----+-----+-----+-----+-----+-----+
| 674988639855312896 | t_order | 2 | 1 | 2021-12-07 14:43:52 | NULL |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.16 sec)
```

• 修改分片规则触发 scaling 迁移

# Database Plus 解决方案-数据安全



# Database Plus 解决方案-数据安全

```
mysql> CREATE ENCRYPT RULE t_encrypt (
->   COLUMNS(
->     (NAME=user_id,PLAIN=user_plain,CIPHER=user_cipher,TYPE(NAME=AES,PROPERTIES('aes-key-value'='123456abc'))),
->     (NAME=order_id,PLAIN=order_plain,CIPHER =order_cipher,TYPE(NAME=RC4,PROPERTIES('rc4-key-value'='123456abc')))
-> );
Query OK, 0 rows affected (0.12 sec)
```

• 创建加密规则和加密表

```
mysql> CREATE TABLE `t_encrypt` (
->   `id` int(11) NOT NULL,
->   `order_id` varchar(45) DEFAULT NULL,
->   `user_id` varchar(45) DEFAULT NULL,
->   PRIMARY KEY (`id`)
-> ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Query OK, 0 rows affected (2.23 sec)
```

```
mysql> INSERT INTO t_encrypt(id, order_id, user_id) VALUES(1, '1234', '2345');
Query OK, 1 row affected (1.43 sec)
```

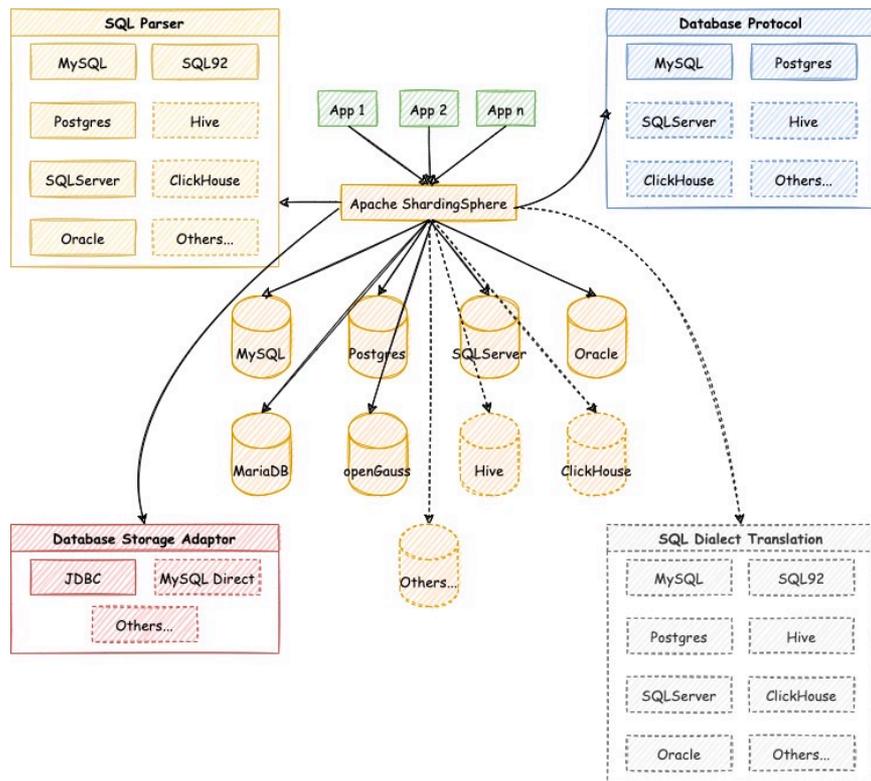
```
mysql> SELECT * FROM t_encrypt WHERE order_id = '1234';
```

id	order_id	user_id
1	1234	2345

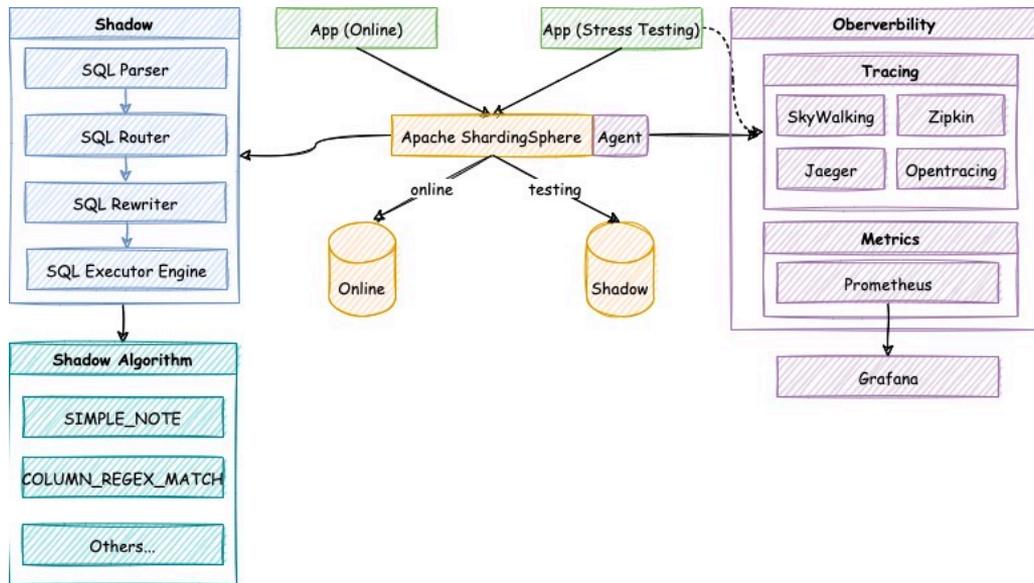
1 row in set (0.16 sec)

• 面向逻辑列编写 SQL, 自动实现加解密

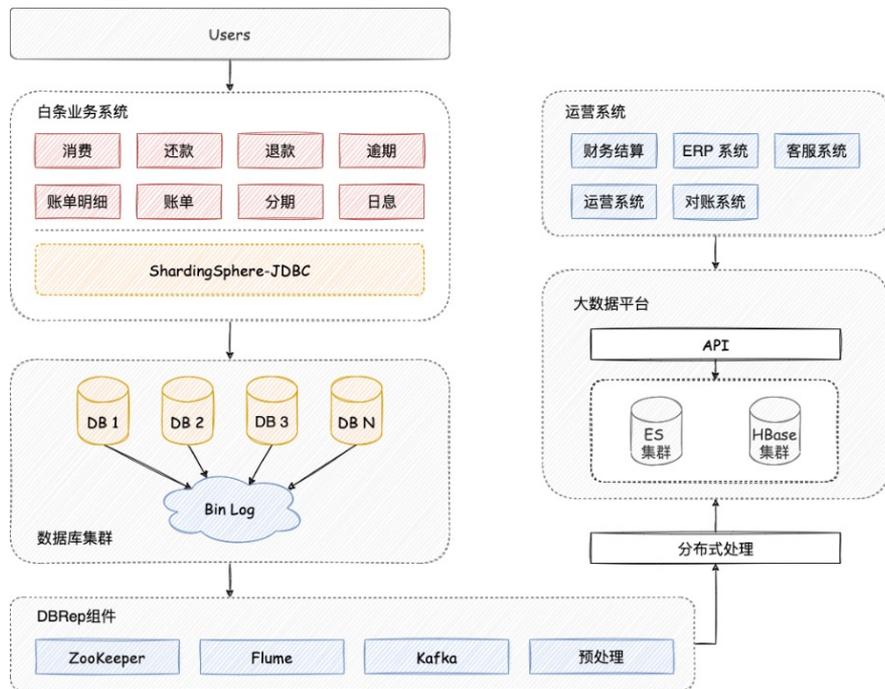
# Database Plus 解决方案-数据库网关



# Database Plus 解决方案-全链路压测



# Database Plus 解决方案-应用案例



京东白条使用 ShardingSphere 解决了**千亿数据存储和扩容**的问题，为大促活动奠定了基础。

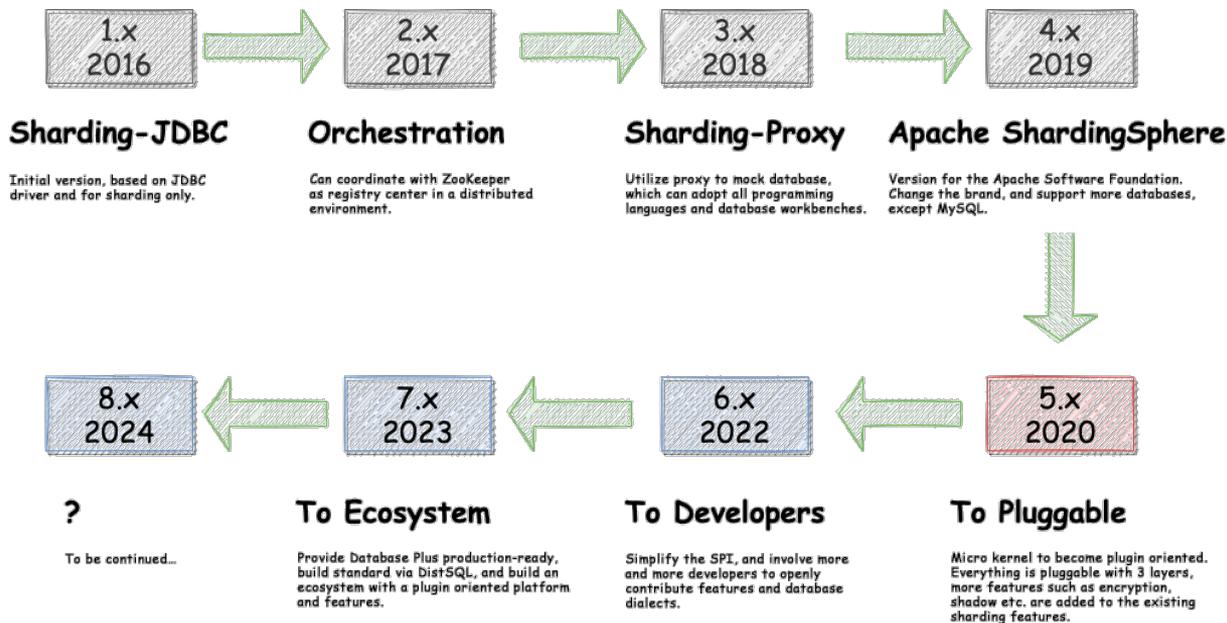
- ◆ **简化升级路径:** 通过架构解耦，业务系统升级所涉及技术栈得到有效速断，研发团队不再需要关注分表设计，精力全部聚焦于业务本身，升级路径得到大幅度优化；
- ◆ **节省研发力量:** 引入成熟的 ShardingSphere 无需重新开发分表组件，在简化业务升级路径的基础上节省了大量研发力量；
- ◆ **架构灵活扩展:** 搭配使用 Scaling 同步迁移组件从容面对 618 和 11.11 等大型活动，系统灵活扩容；

# 目录

---

1. ShardingSphere 简介
2. ShardingSphere Database Plus 架构解读
3. ShardingSphere Database Plus 解决方案
4. ShardingSphere 未来规划

# ShardingSphere 未来规划



# ShardingSphere 未来规划

项目地址: <https://shardingsphere.apache.org>

GitHub 地址: <https://github.com/apache/shardingsphere>

开发者邮件列表: [dev-subscribe@shardingsphere.apache.org](mailto:dev-subscribe@shardingsphere.apache.org)

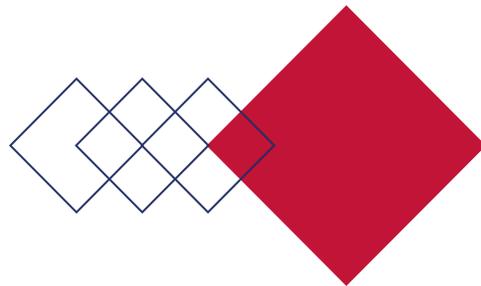
中文社区: <https://community.sphere-ex.com>



技术干货



技术交流



# Thanks

