



开源软件供应链点亮计划—开源之夏2023

Apache ShardingSphere 开源之夏宣讲介绍

陈出新

tuichenchuxin@apache.org

ISCAS 中国科学院软件研究所
Institute of Software Chinese Academy of Sciences

 **OpenEuler**



个人简介



陈出新

SphereEx 研发工程师

Apache ShardingSphere Committer

- 负责 ShardingSphere 内核模块相关功能的研发工作，在分布式数据库领域，对数据分片、加密、调度、查询优化等方面有着丰富的专业知识和实践经验。

01

Apache ShardingSphere 简介

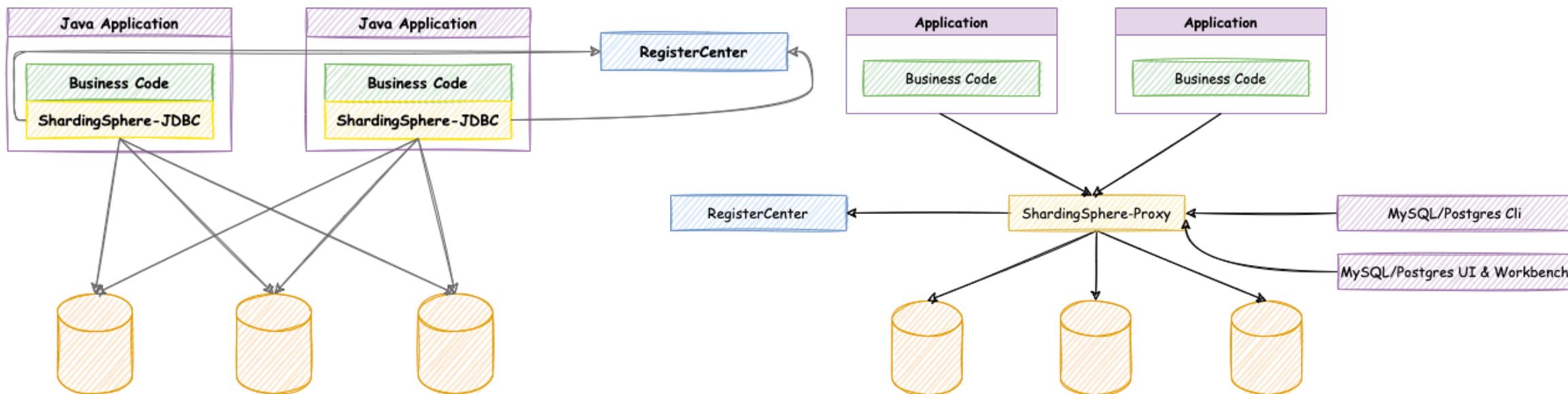
ShardingSphere 简介

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」**。

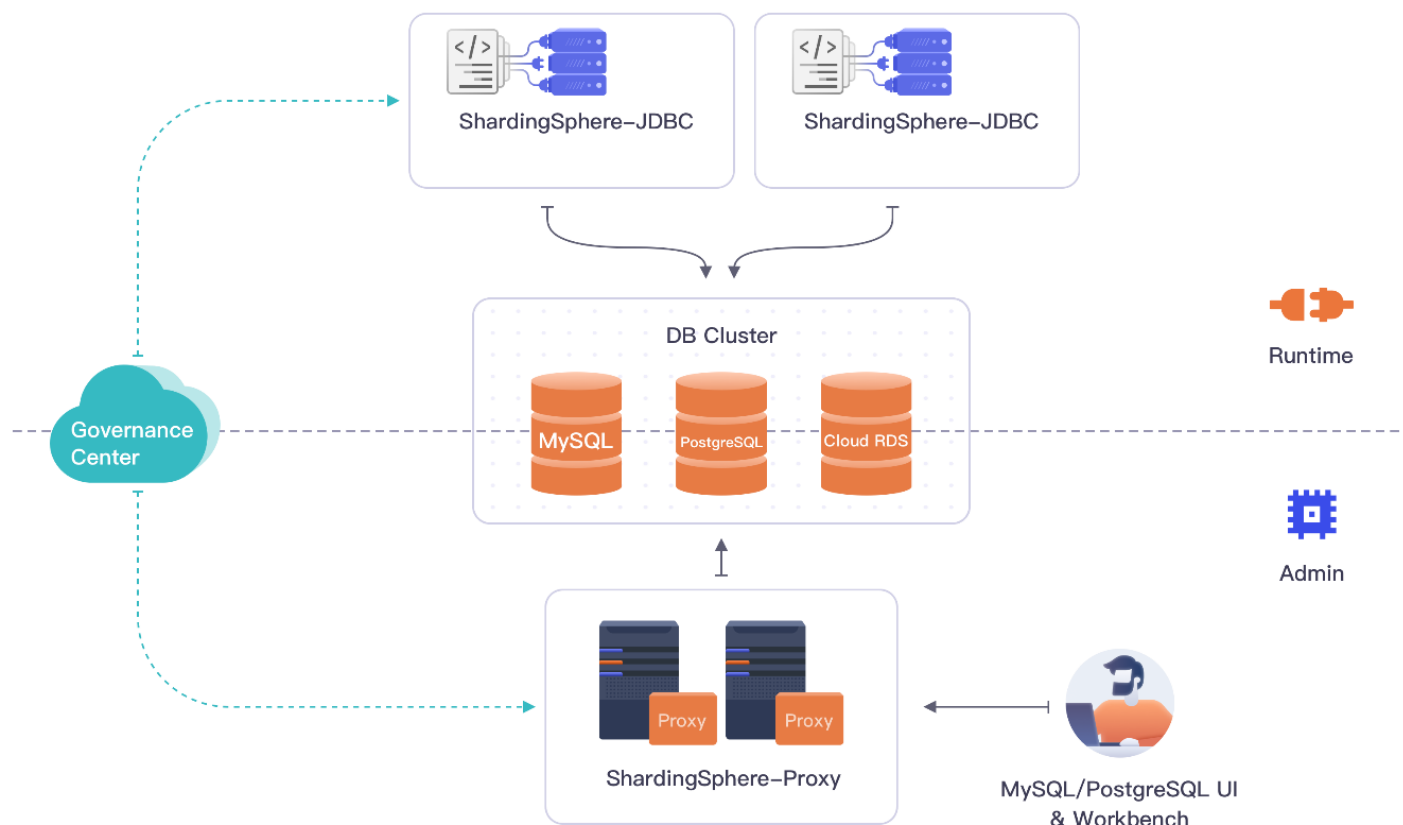


ShardingSphere 接入端



- ◆ ShardingSphere 由 JDBC 和 Proxy 2 个接入端组成，既支持「独立部署」，也可「混合部署」；
- ◆ JDBC 和 Proxy 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 等各种应用场景；

ShardingSphere 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。

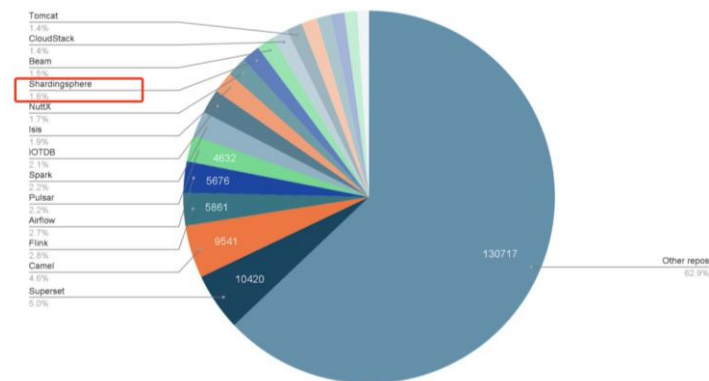
ShardingSphere 社区



- ◆ 全球最顶级的开源软件基金会
- ◆ 管理超过两亿行代码
- ◆ 成功孵化 300+ 顶级开源项目



Projects by Number of Commits, 2021



- **18,000+** Stars
- **16,000+** Pull Requests
- **6,000+** Forks
- **500+** Contributors

基金会认可 Apache 软件基金会顶级项目

社区活跃度 2021 年度 Apache 基金会年度报告代码提交量位列前十

学术界认可 数据库顶会 ICDE 2022 发表论文《A Holistic and Pluggable Platform for Data Sharding》



出版《**Apache ShardingSphere 的权威指南**》，有力阐述了如何在多模型数据库之上构建开放生态

02

Apache ShardingSphere 设计理念 & 基本原理

ShardingSphere 设计理念

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。

ShardingSphere 技术架构

Database Plus

构建多模数据库上层的标准和生态

连接

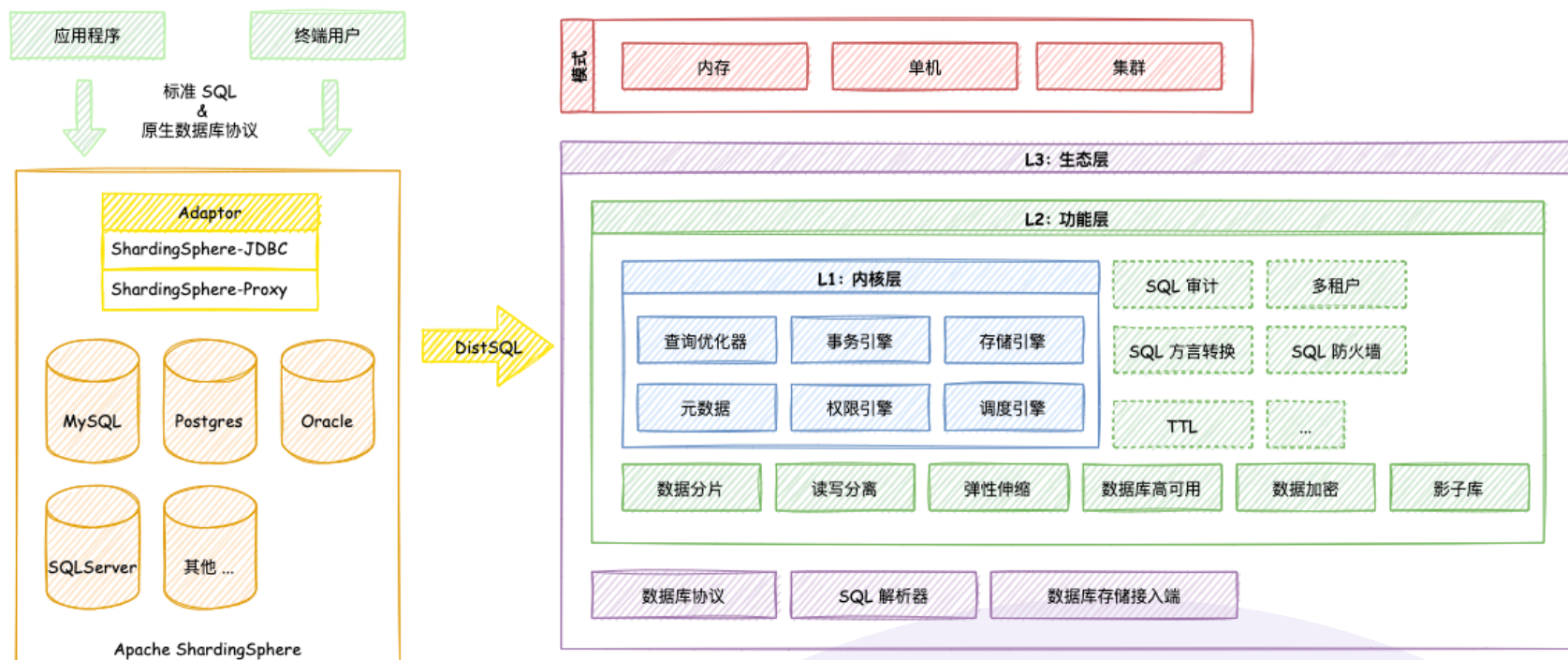
连接数据和应用，关注多模数据库之间的合作

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

微内核全面面向可插拔的三层架构体系设计

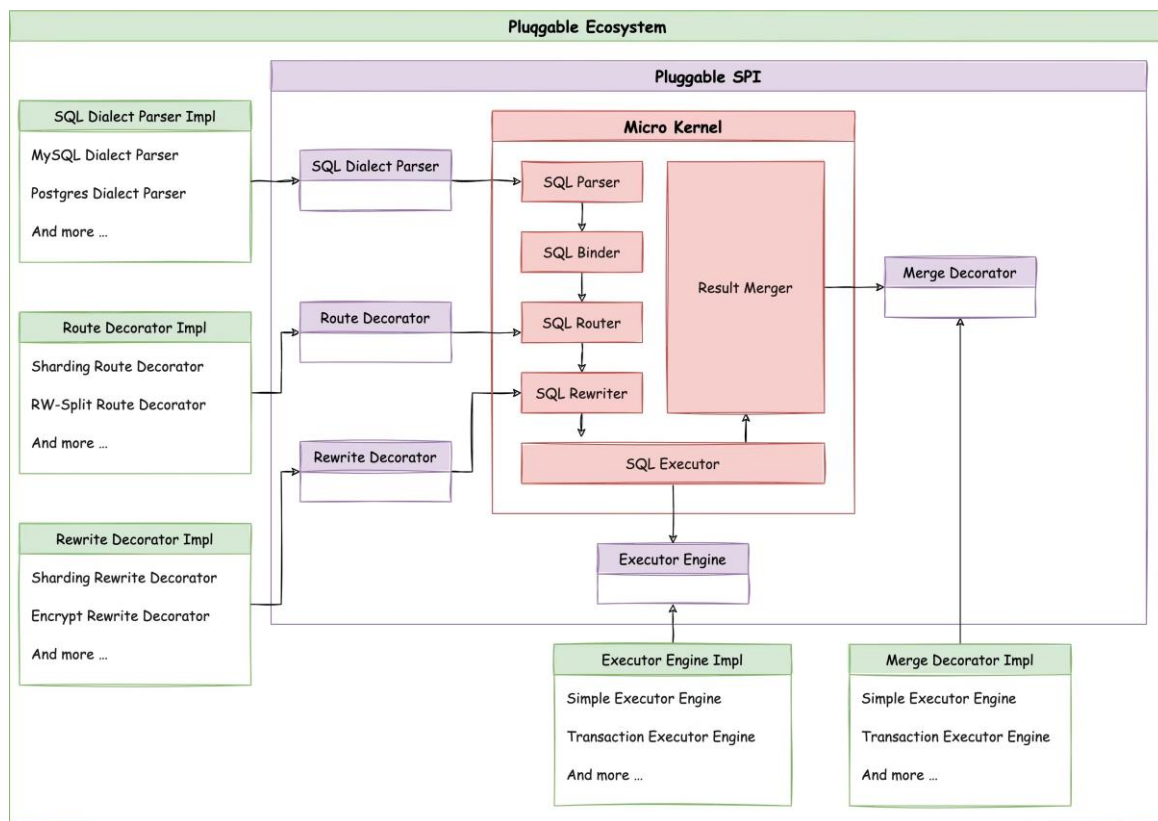


- ◆ L1 内核层：面向数据库内核，包括数据库事务引擎，查询优化器等；

- ◆ 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；

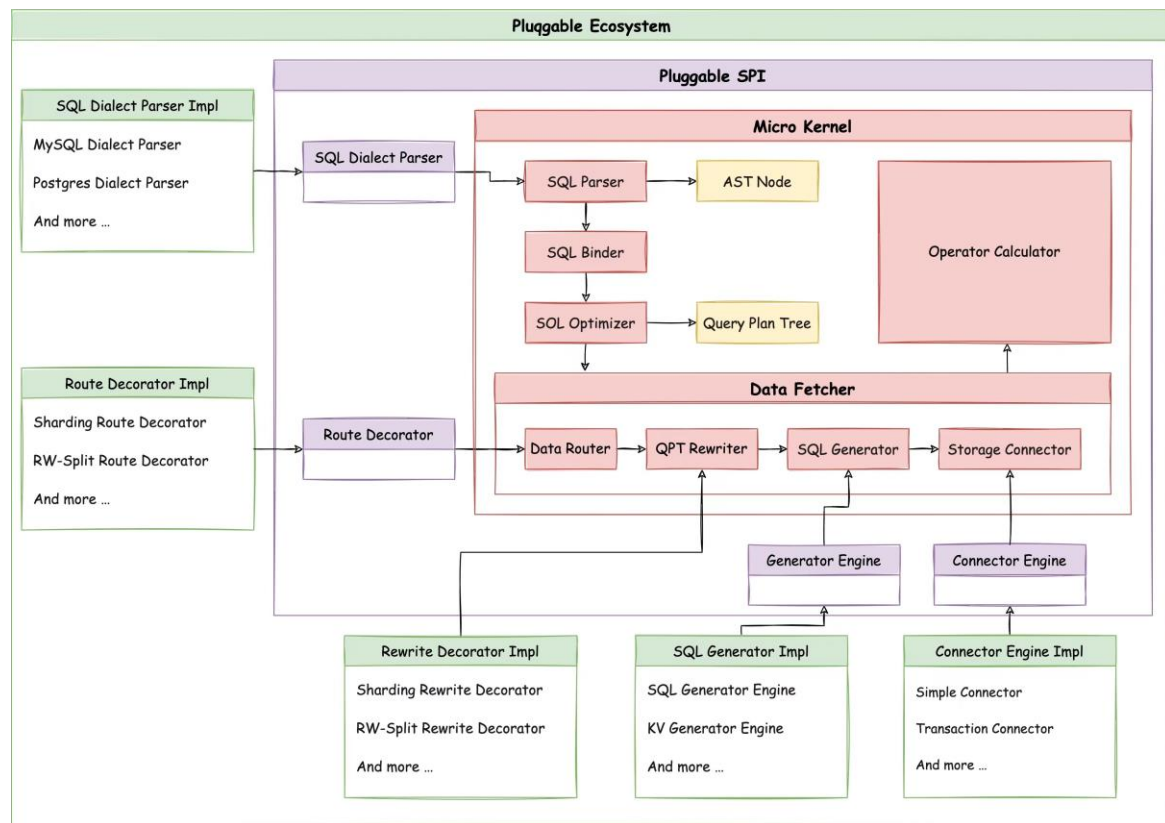
- ◆ L3 生态层：通过三个接口分别实现数据库协议、SQL 方言和数据库存储对接，用于打造异构数据网关；

ShardingSphere 微内核 - 计算下推引擎



- ◆ 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、数据脱敏等功能；

ShardingSphere 微内核 - 联邦查询引擎



- ◆ **SQL Optimizer**：对跨库的关联查询及子查询进行 RBO 和 CBO 优化，得到最优执行计划；
- ◆ **Data Fetcher**：根据最优执行计划生成的 SQL 从存储节点获取数据，Data Fetcher 内部包含了对生成 SQL 的路由、改写和执行；
- ◆ **Operator Calculator**：根据最优执行计划以及从存储节点获取的数据进行算子计算，得到最终查询结果。

ShardingSphere 解决方案

基于 ShardingSphere 的可插拔架构底座，将分布式数据库的核心能力中，与企业业务贴近的垂直功能排列组合，在数据库替换，海量数据存储，高并发访问与数据安全存储等场景，提供对应的解决方案。



面向分布式的解决方案

分布式数据库解决方案



面向数据控制的解决方案

数据安全解决方案



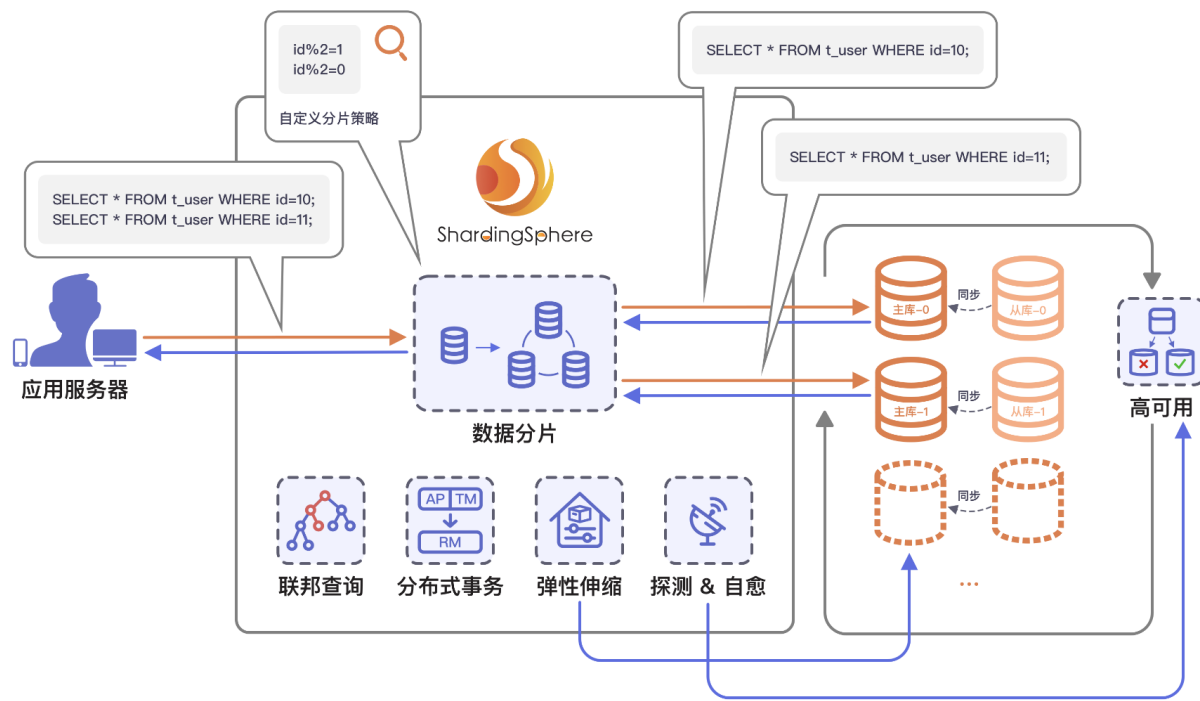
面向流量控制的解决方案

全链路压测解决方案

ShardingSphere 解决方案 – 分布式数据库

分布式数据库解决方案：

为解决原有方案的技术瓶颈，降低更换架构带来的复杂性风险，在不更换原有架构前提下，实现数据库同步，管理多个异构数据库集群，线性提升数据存储容量及并发吞吐。为用户提供基于数据分片，分布式事务、弹性伸缩的分布式数据库解决方案，兼具单机交易型数据库稳定性和分布式数据库的扩展能力。

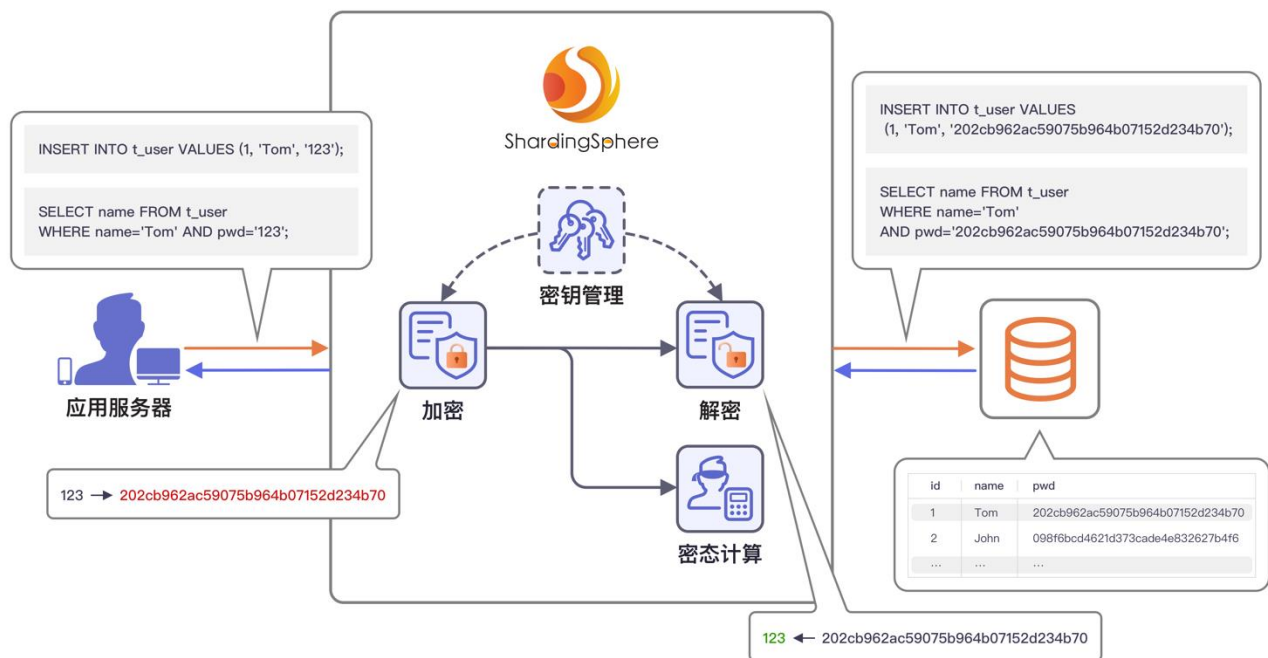


功能特性

- 极致性能 & 多接入端
- 多数据库方言 & 访问协议支持
- 水平拆分 / 内置 & 自定义算法
- 垂直拆分 / 联邦查询
- 分布式事务 / 多类型灵活切换
- 弹性伸缩 / 数据迁移
- 读写分离 / 高可用
- 分布式管控 / 监控

ShardingSphere 解决方案 – 数据安全

数据安全解决方案： 为了防止数据泄露，保护数据安全，提供基于产品的数据加密和数据脱敏功能。在不改动原有代码的前提下，为企业提供跨平台、异构环境的数据安全解决方案。

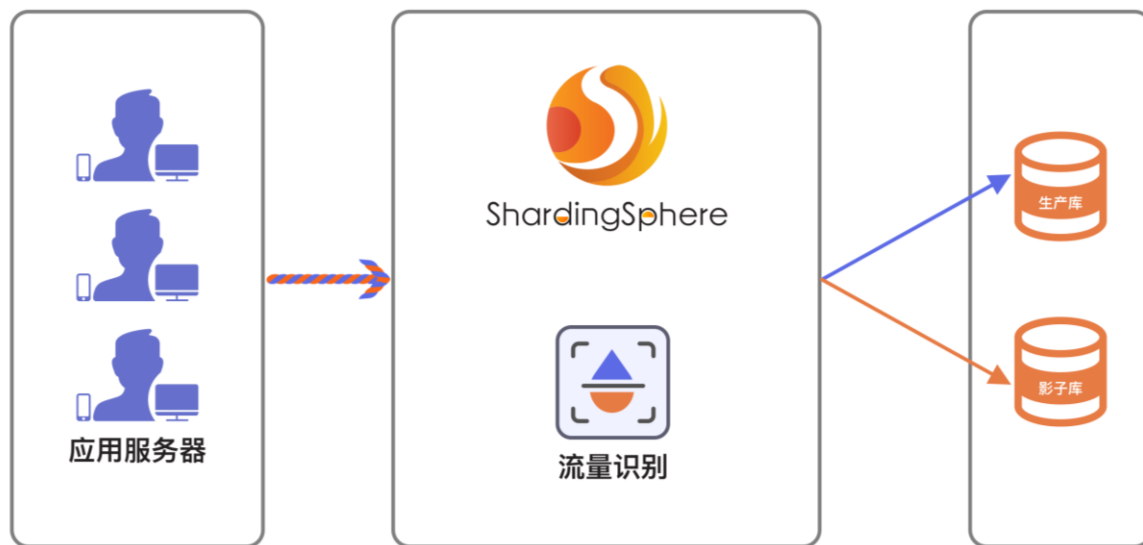


功能特性

- 极致性能 & 多接入端
- 多数据库方言 & 访问协议支持
- 数据加密 / 国密支持
- 密态计算
- 在线加密 / 转加密
- 权限控制 / 对接 LDAP & 第三方

ShardingSphere 解决方案 – 全链路压测

全链路压测解决方案：基于内核的 SQL 解析能力，以及可插拔平台架构，实现压测数据与生产数据的隔离，帮助应用自动路由，支持全链路压测。帮助用户实现在生产环境进行压测，所获得较为准确地反应系统真实容量水平和性能的测试结果。



→ 代表压测流量

→ 代表线上正常流量

功能特性

- 极致性能 & 多接入端
- 多数据库方言 & 访问协议支持
- 内置 & 自定义影子库算法
- 可观察性 / Metric & Tracing
- 金丝雀测试
- 灰度上线

04

开源之夏任务介绍

增强 Oracle、PostgreSQL 语法兼容性（基础 2 个任务）

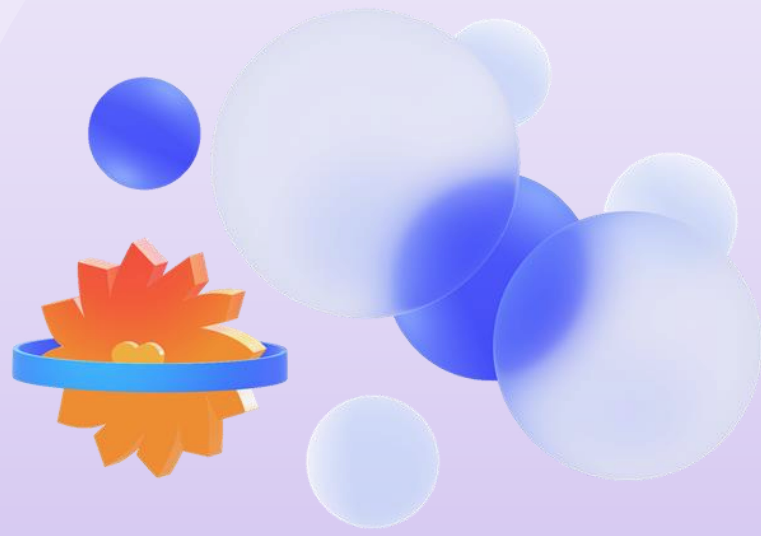
背景：

ShardingSphere Parser Engine 帮助用户将 SQL 语句解析为抽象语法树，并从语法树生成对应的 SQL Statement 对象。Parser Engine 目前支持 `MySQL`、`PostgreSQL`、`SQLServer`、`openGauss` 和 `Oracle` 等多种数据库方言，为了提升对于不同数据库方言的支持度，我们需要对 Parser Engine 中未支持的 SQL 进行优化。更多关于 Parser Engine 的介绍请参考：

<https://shardingsphere.apache.org/document/current/en/reference/sharding/parse/>。

相关技能：

1. 精通 java
2. 理解 Antlr4 语法
3. 熟悉 Oracle、PostgreSQL 数据库



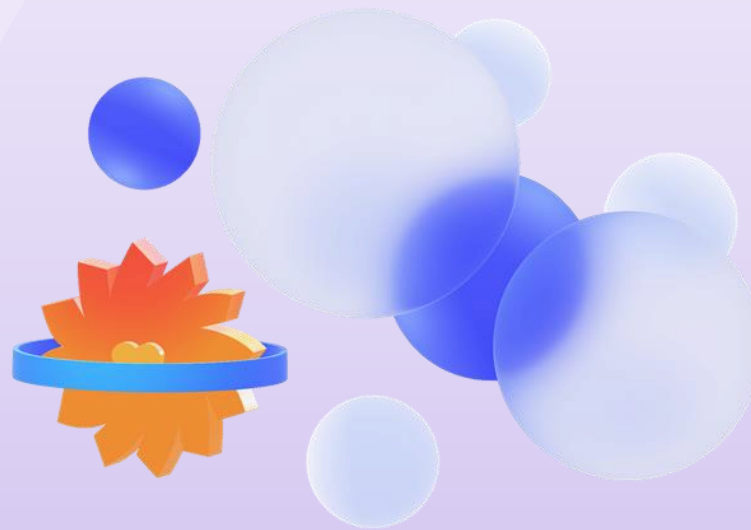
增强 SQLNodeConverterEngine，支持更多的 MySQL/PostgreSQL/ opengauss SQL 语句（进阶）

背景：

ShardingSphere SQL federation 引擎提供了对复杂 SQL 语句的支持，它可以很好地支持跨数据库连接查询、子查询、聚合查询和其他语句。SQL federation 引擎的一个重要组成部分就是将 ShardingSphere 解析的 SQL 语句转换为 SqlNode，利用 Calcite 实现 SQL 优化和联邦查询。

相关技能：

1. 掌握 Java 语言
2. 熟悉 Antlr g4 语法编写
3. 熟悉 MySQL、PostgreSQL 和 openGauss 数据库，以及 Calcite 中的 SqlNode



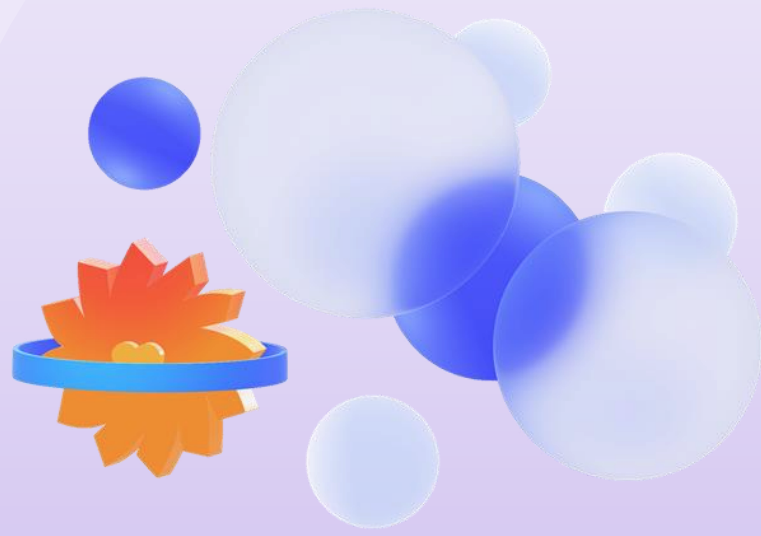
Apache ShardingSphere E2E-SQL 使用 Awaitility 异步框架代替 Sleep 逻辑（基础）

背景：

Apache ShardingSphere 的集成测试涵盖了监控、迁移、事务、DistSQL，以及其它数据库方言语句。现在各个集成测试的模块都包含了大量的 `Thread.sleep(x)` 逻辑，它存在的主要目的是等待元数据内存刷新的完成。这样虽然可以达到我们测试的目的，但其实有些时候根本不需要等待所设置的 sleep 时间即可完成内存的元数据刷新，并且这样是影响我们 Github Action 整体的 CI 时间的，所以我们打算引入 Awaitility 框架来替换 `Thread.sleep(x)`，这样可以间接的提高我们的开发效率。

相关技能：

1. 掌握 Java 语言
2. 熟悉 e2e-sql 项目并了解它的运行过程
2. 熟悉 Awaitility 异步框架的使用
3. 熟悉 Apache ShardingSphere DistSQL



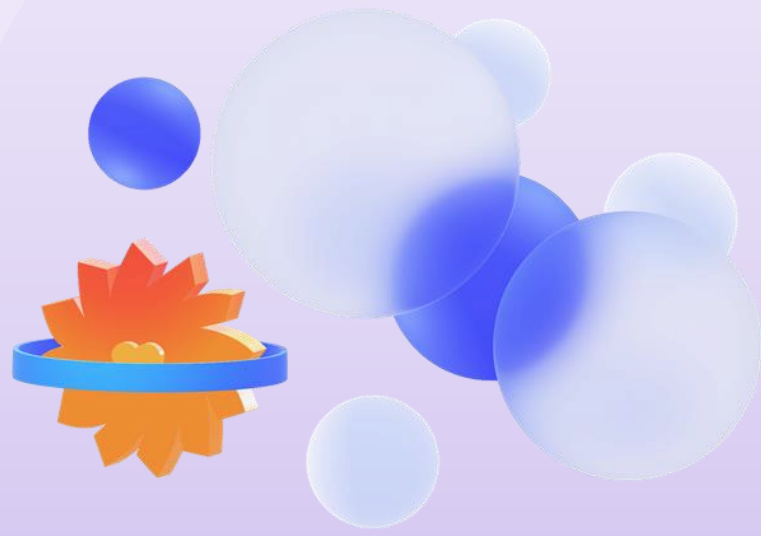
Apache ShardingSphere: Search Path (进阶)

背景：

Apache ShardingSphere 的定位是 Database Plus，目标是在异构数据库之上构建一个标准层和生态系统，所以 ShardingSphere 使用的存储节点可以是 PostgreSQL / openGauss。当用户使用 PG / OG 时，可能提前在系统变量中设置了查询 schema 的优先级，比如当用户在执行 SQL: `SELECT * FROM order;` 语句时，若此时未指定 schema。则在 PG / OG 上执行逻辑为：按照 `search_path` 的 schema 逐一进行扫描，直至命中 `order` 表。在 ShardingSphere 上执行逻辑：目前一律查询 `public` schema。所以打算在 ShardingSphere 中也支持 PG / OG 的 `search_path` 功能，提升用户体验。

相关技能：

- 1.精通 Java
- 2.熟悉 PostgreSQL / openGauss 数据库
- 3.熟悉 ShardingSphere





开源软件供应链点亮计划—开源之夏2023

谢谢观看

Apache ShardingSphere 官方中文社区: <https://community.sphere-ex.com>

Apache ShardingSphere Website: <https://shardingsphere.apache.org>

Apache ShardingSphere GitHub: <https://github.com/apache/shardingsphere>

Apache ShardingSphere Slack Channel: <https://apacheshardingsphere.slack.com>



Apache ShardingSphere
微信公众号

