

Apache ShardingSphere Shadow DB 及在 CyborgFlow 中的应用

侯阳

2022年1月15日



SphereEx 中间件研发工程师

Apache ShardingSphere Contributor

目前专注于 Shadow DB 及全链路压测相关的设计和研发





1. Apache ShardingSphere 整体的架构设计

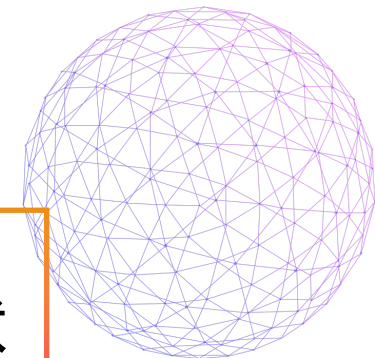


2. Apache ShardingSphere Shadow DB 的架构设计与应用场景



3. CyborgFlow 中 Shadow DB 的数据路由能力



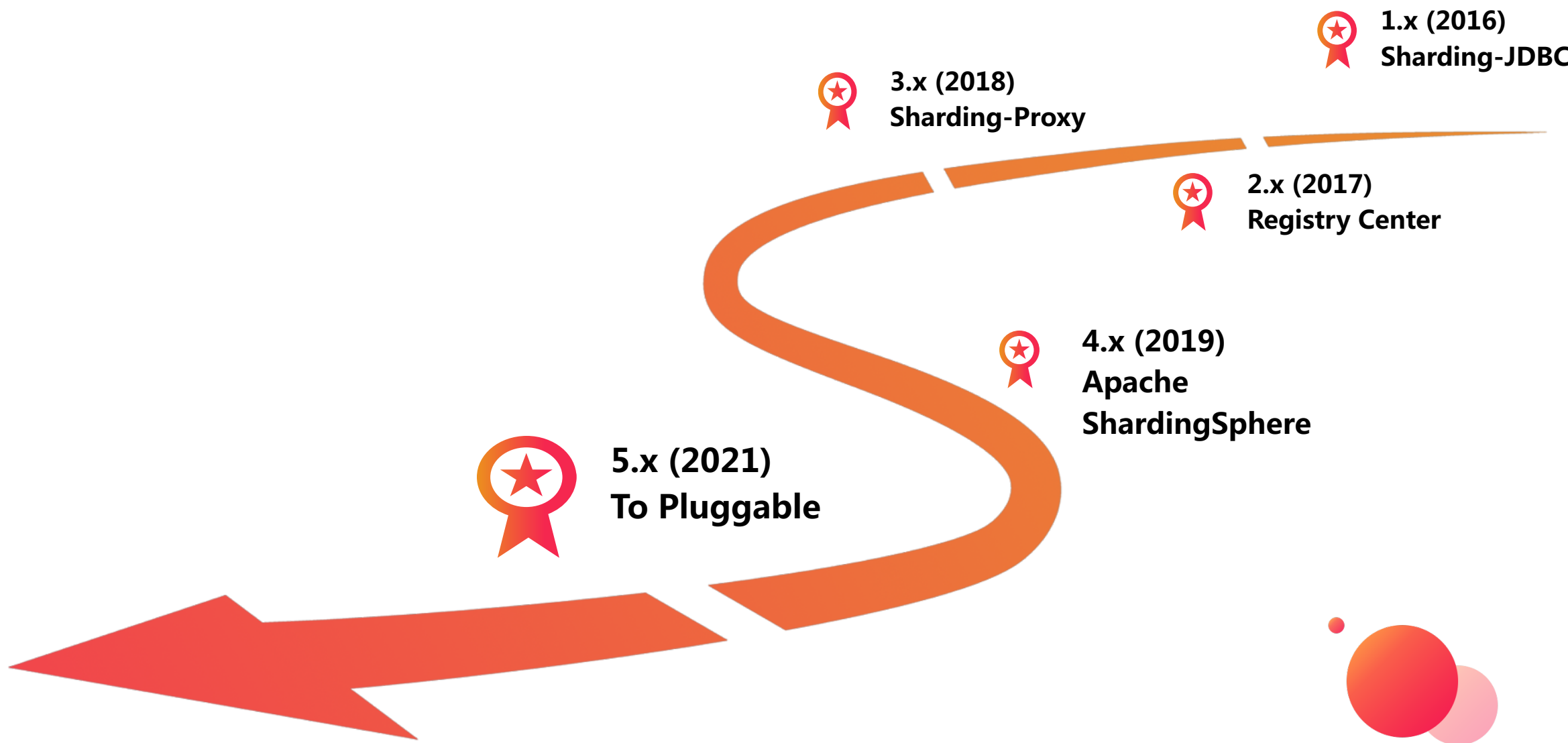


Apache ShardingSphere 发展和现状

Apache ShardingSphere 架构设计

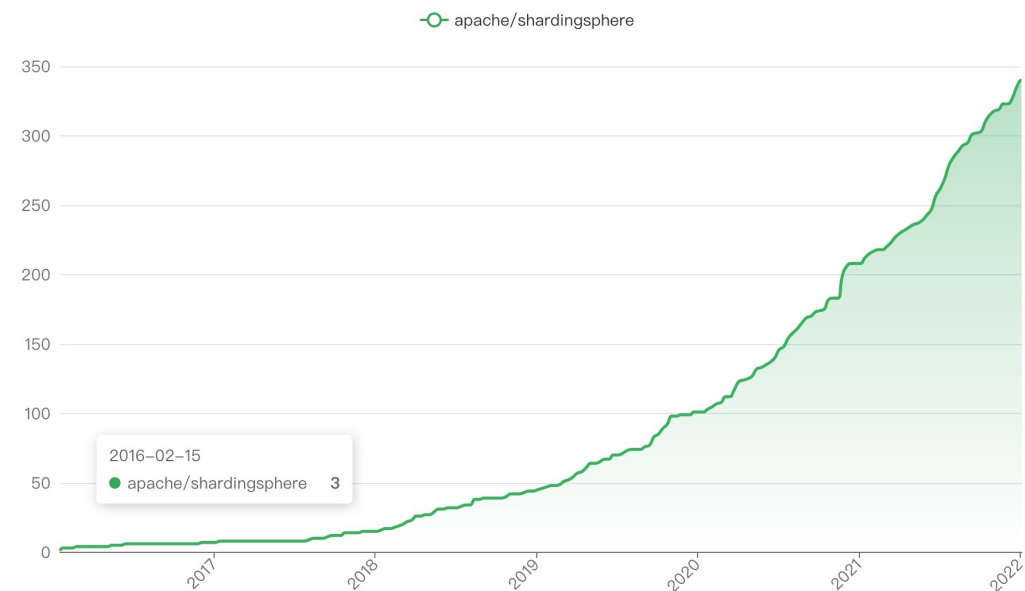
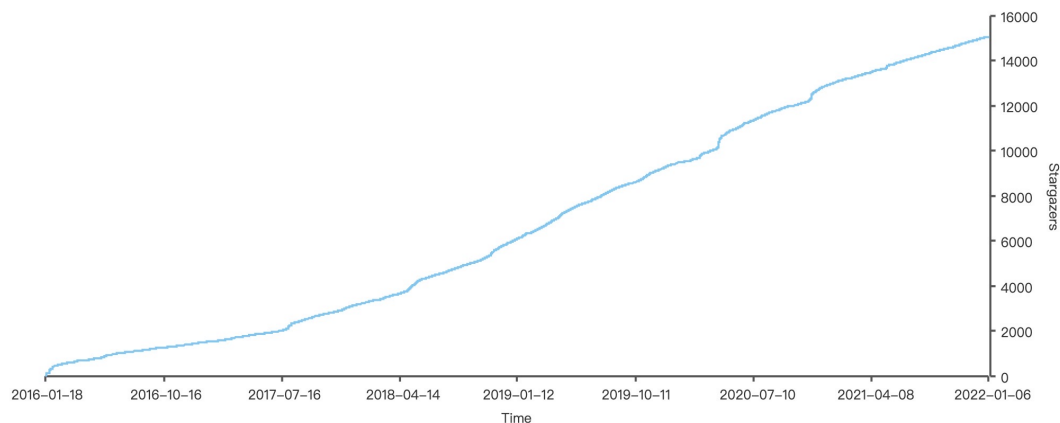


Apache ShardingSphere Roadmap



Apache ShardingSphere Status

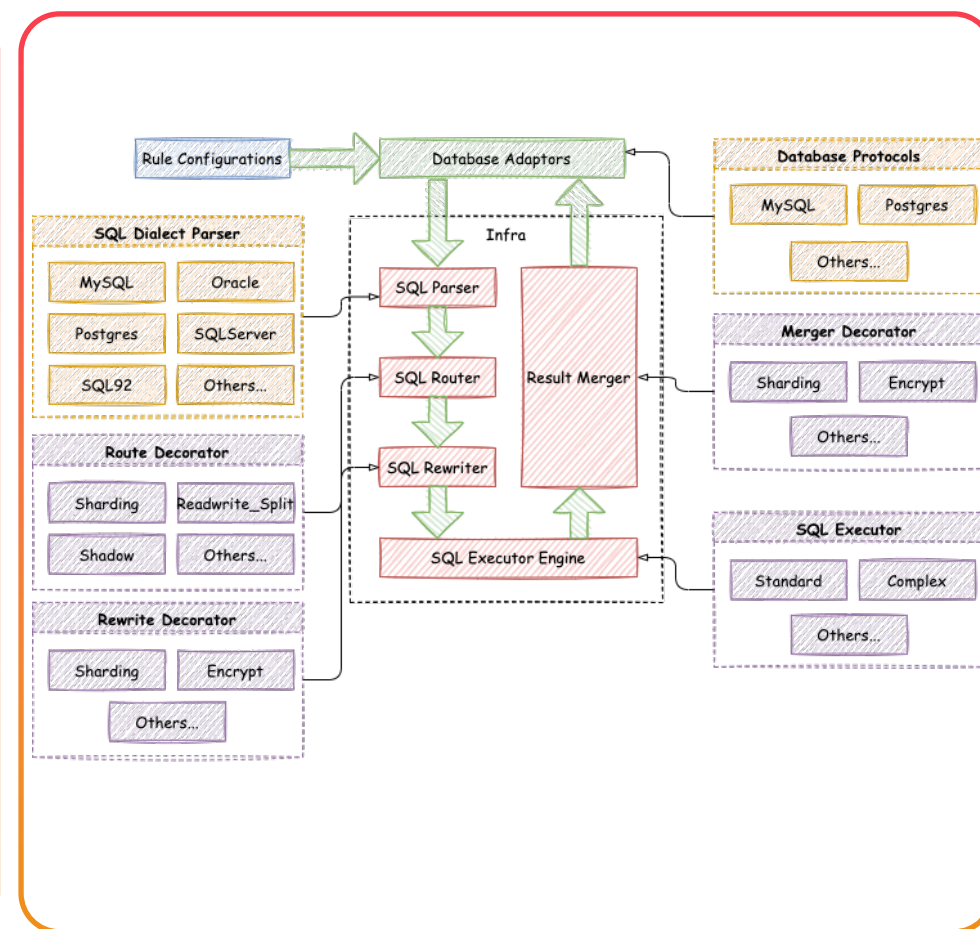
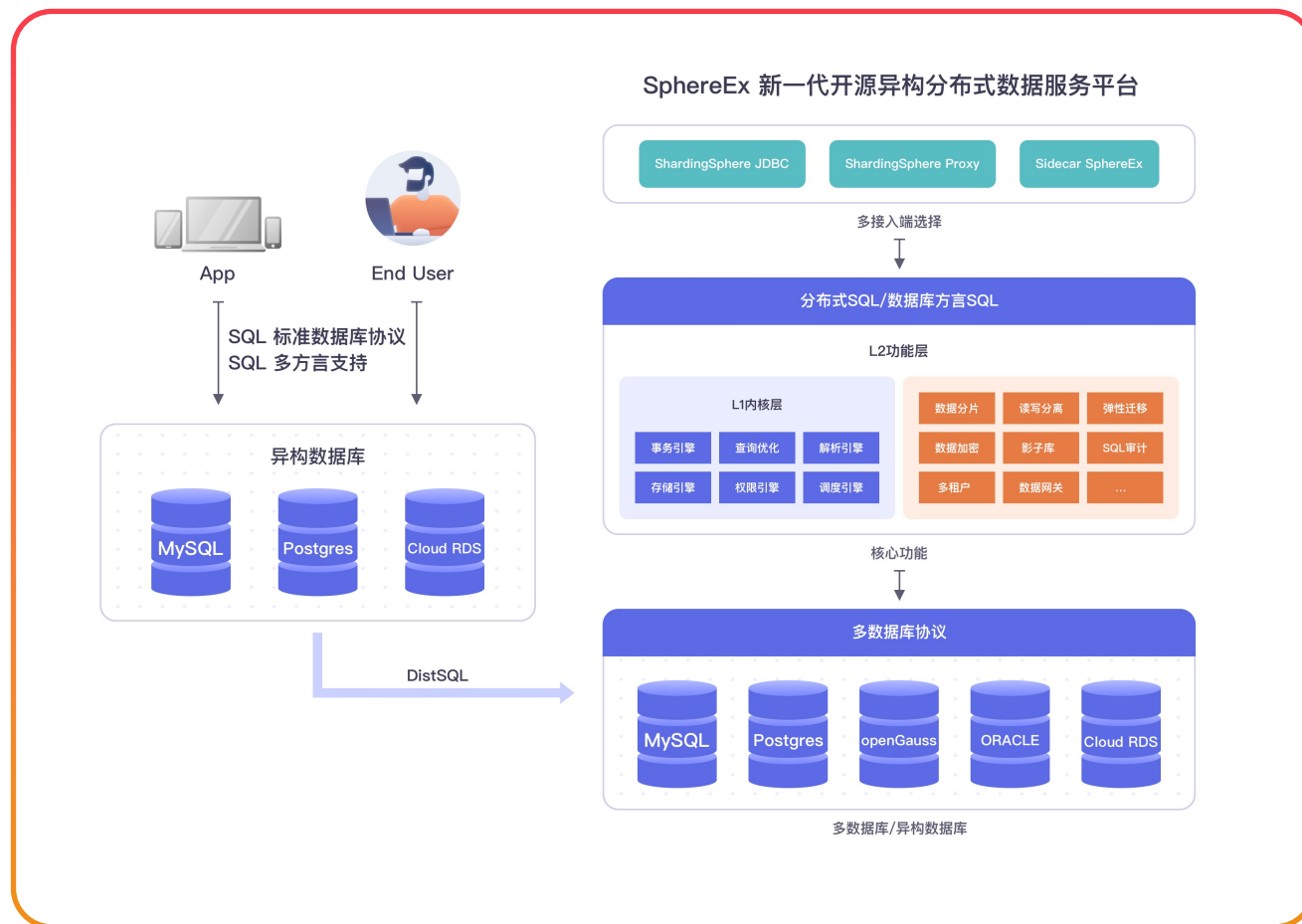
Awesome! [apache/shardingsphere](https://github.com/apache/shardingsphere) was created 5 years ago and now has **15073** stars.

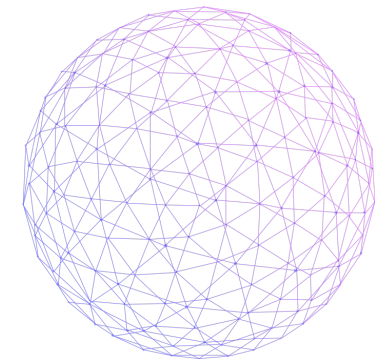


GitHub: <https://github.com/apache/shardingsphere>

OpenSEC: <https://community.sphere-ex.com>

Apache ShardingSphere Overview





Shadow DB 应用场景

Shadow DB 架构设计



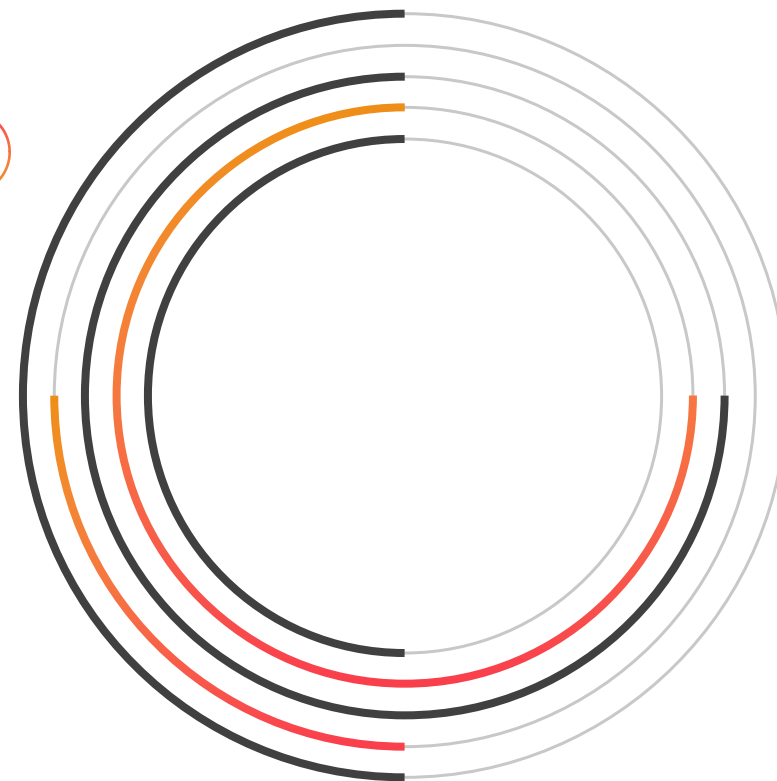
Shadow DB Scenes

全链路压测：流量规划，比如双 11，618 等

灰度发布：指定测试用户的体验版本发布

服务预热：预热数据隔离

对比测试：基于版本的对比测试



Shadow DB API

4.1.1

rules:

- !SHADOW

column:

shadowMappings:

ds: shadow_ds

Shadow DB 重构的原因



01. 业务 SQL 改造

用户需要根据实际的业务需求，对压测相关 SQL 进行相应改造，测试完成还要修改回来。



02. SQL 改写性能损耗

执行 SQL 改写操作会增加损耗，降低压测结果的准确性。

Shadow DB API

5.0.0

```
rules:
- !SHADOW
  enable: true
  dataSources:
    <shadow-data-source>:
      sourceDataSourceName: ds
      shadowDataSourceName: shadow_ds
  tables:
    <shadow-table-name>:
      dataSourceNames:
        - <shadow-data-source>
      shadowAlgorithmNames:
        - <shadow-algorithm-name>
  shadowAlgorithms:
    <shadow-algorithm-name>:
      type:
      props:
        xxx: xxx
```

Shadow DB 重构后提升



01. 无需业务 SQL 改造

用户无需根据实际的业务需求，对压测相关 SQL 进行改造，只需要控制测试数据就可以进行测试。



02. 无需 SQL 改写提升性能

执行 SQL 移除了改写操作降低了损毁，提升压测结果的准确性。

Shadow DB Concept



Shadow Datasource

影子数据源

用于存储隔离数据的数据源。

压力测试尽量与生产数据库一致。

功能测试和灰度测试可以使用相同数据库。



Shadow Table

影子表

压测业务涉及相关的表我们称为影子表。压测相关业务场景需要很多基础信息比如字典表等，并不会产生压测数据。影子库只需要创建产生压测数据的影子表即可。



Shadow Algorithm

影子算法

影子算法是判定执行 SQL 是否执行到影子库的判定算法。

一个影子表可以关联多个影子算法。只要有一个影子算法判定成功执行 SQL 就会路由到影子库。



Shadow Algorithm



Column Value Match

shadowAlgorithms:

<algorithm-name>:

type: VALUE_MATCH

props:

operation: insert

column: user_id

value: 1

性能损耗非常小，最差情况
3% 左右



Column Regex Match

shadowAlgorithms:

<algorithm-name>:

type: REGEX_MATCH

props:

operation: insert

column: user_id

regex: "[1]"

匹配灵活性度高



Simple Hint

shadowAlgorithms:

<algorithm-name>:

type: SIMPLE_HINT

props:

foo: bar

SQL 支持度高



Shadow DB Routing Process

Shadow Table



影子表判定

判定 SQL 执行语句涉及的表与配置的影子表是否有交集

Shadow Algorithm



影子算法判定

依次判定影子表下配置的影子算法，有任何算法判定成功。影子算法判定成功。

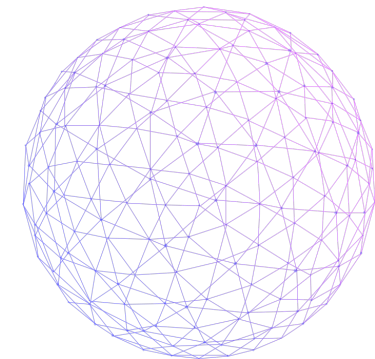
Shadow Data Source Router



数据源路由

根据配置的影子数据源映射对执行的数据源进行改写。执行 SQL 会路由到配置的影子数据源中执行。





CyborgFlow 中的 Shadow DB

cyborg-database-shadow 演示



CyborgFlow Modules



gateway

cyborg-flow-gateway :
由 Apache APISIX 提供并维护。



agent & dashboard

cyborg-agent &
cyborg-dashboard :
由 Apache SkyWalking
提供并维护。



database-shadow

cyborg-database-shadow :
由 Apache
ShardingSphere 提供并维护。



CyborgFlow Process

cyborg-flow-gateway



网关标记请求

```
routes:
-
  uri: /order/create
  plugins:
    proxy-rewrite:
      headers:
        sw8-correlation: Y3lib3JnLWZsb3c=:dHJ1ZQ==
    skywalking:
      sample_ratio: 1
  upstream:
    nodes:
      "127.0.0.1:8082": 1
    type: roundrobin
```

sw8-correlation: cyborg-flow=true

cyborg-agent



通过 agent 在调用链传递

启动项目时使用 cyborg-agent

启动参数：

- javaagent:{path}/

cyborg-agent/skywalking-agent.jar

网关添加的标识会在 jdbc 调用时会以注解的方式添加到执行 SQL 尾部

例如：

select * from table /*cyborg-flow:true*/

cyborg-database-shadow



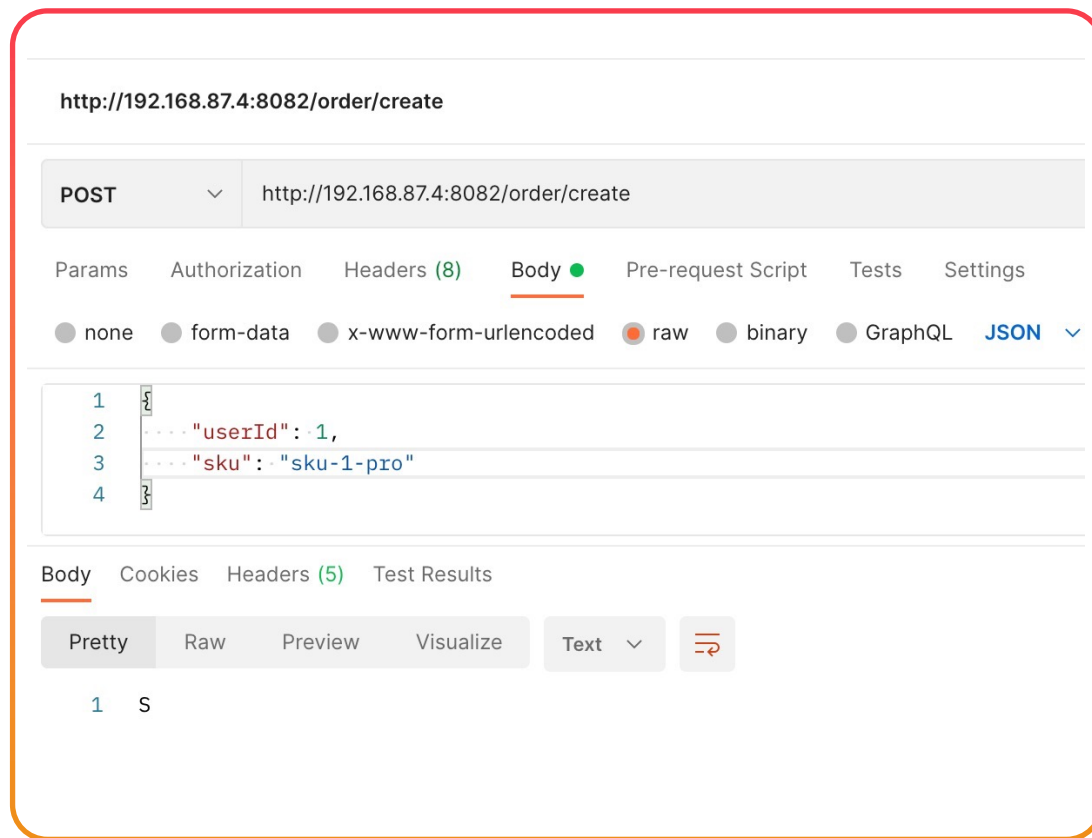
通过 database-shadow 数据路由

```
rules:
- !SHADOW
  enable: true
  dataSources:
    shadowDataSource:
      sourceDataSourceName: ds
      shadowDataSourceName: ds_shadow
  defaultShadowAlgorithmName: simple-note-algorithm
  shadowAlgorithms:
    simple-note-algorithm:
      type: SIMPLE_NOTE
      props:
        cyborg-flow: true
```

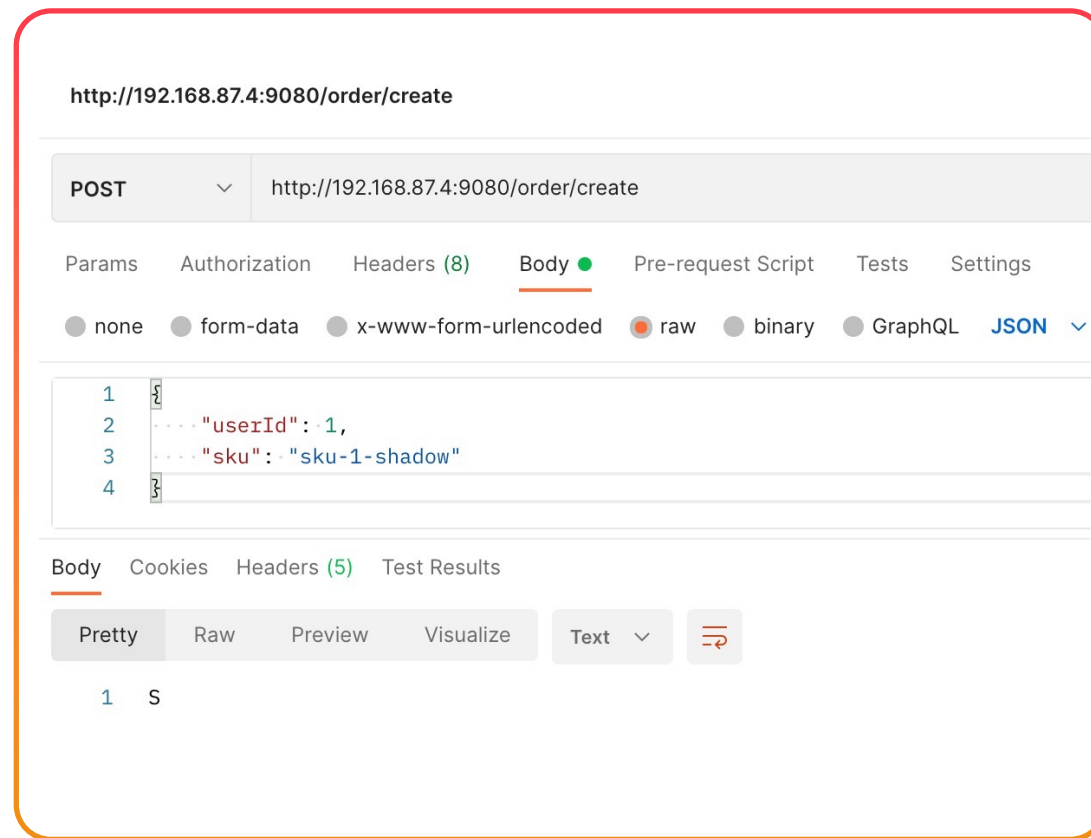
根据配置解析出 执行 SQL 注解中的
/*cyborg-flow:true*/ 匹配成功过行到影子库



CyborgFlow POC1

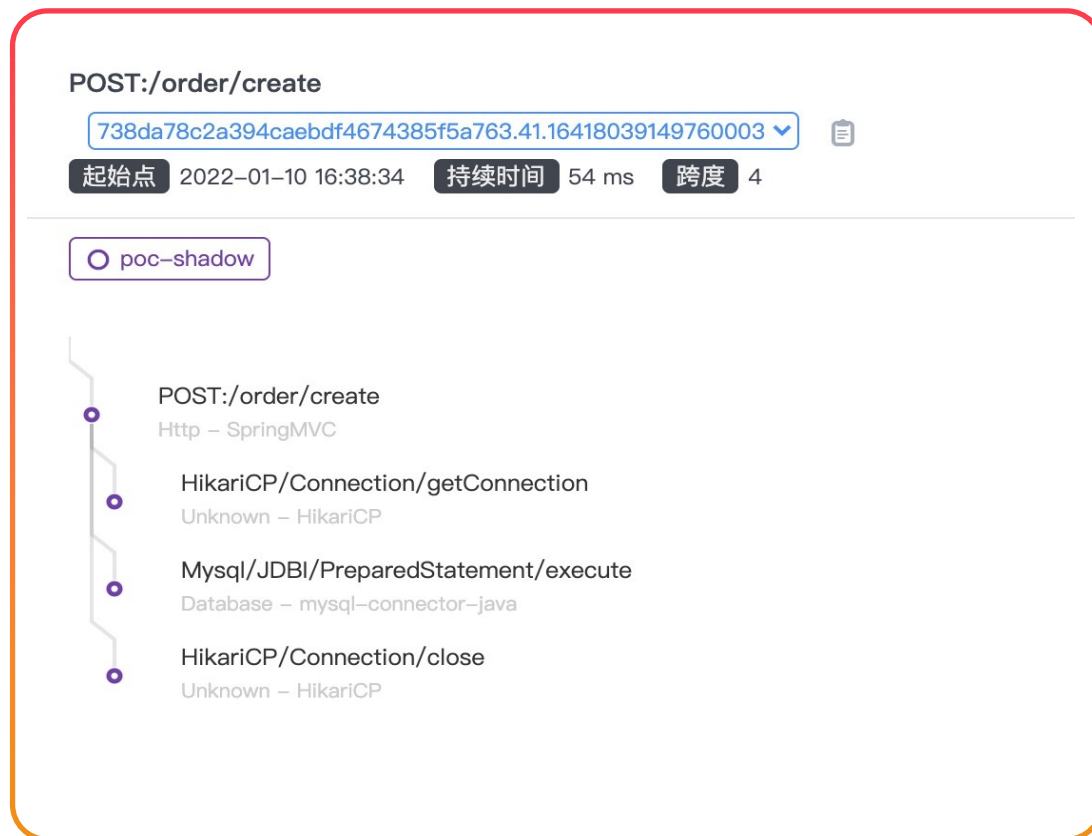


直接调用服务应用模拟生产环境调用

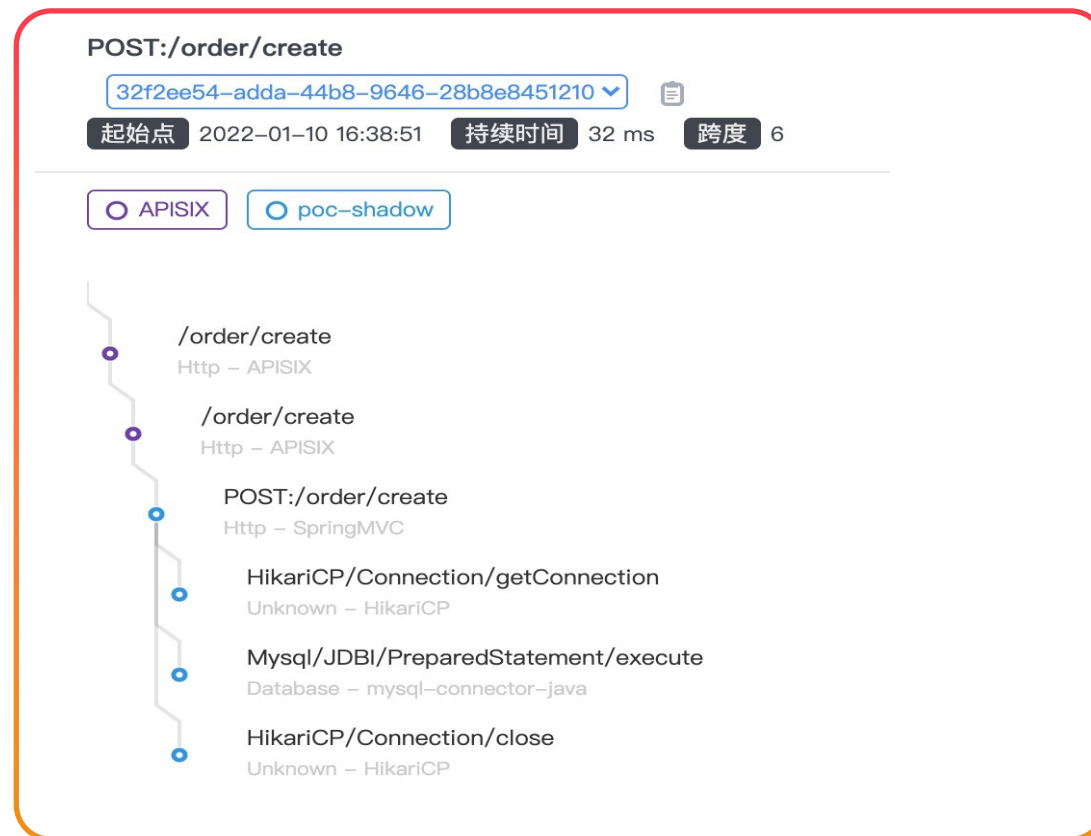


调用 cyborg-flow-gateway 网关模拟压测环境调用

CyborgFlow POC2



真实流量的调用链路



压测流量的调用链路

CyborgFlow POC3

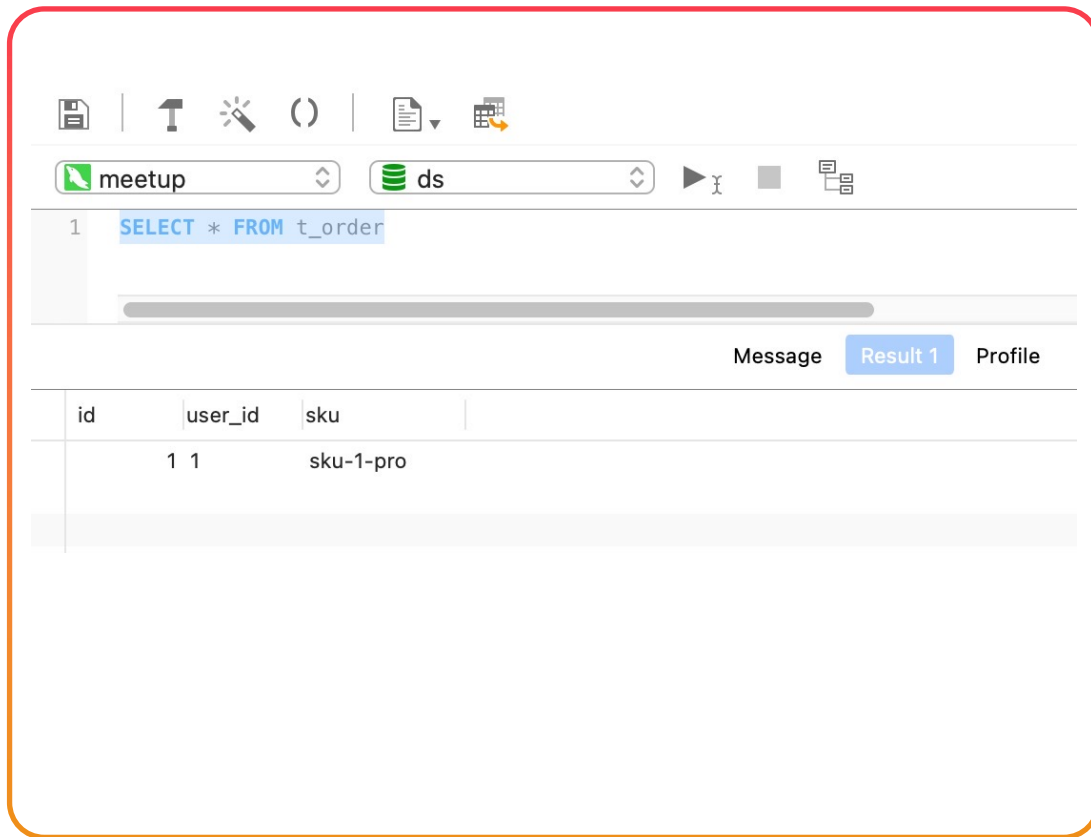
```
poc-shadow
c2497d03776d4cdf9665cd4f19926c31@172.17.0.1
Mysql/JDBC/PreparedStatement/execute
Exit
mysql-connector-java
localhost:3307
false
sql
cyborg-database-shadow
INSERT INTO t_order (user_id, sku) VALUES (?, ?) 
```

真实流量的调用 SQL 详情

```
poc-shadow
c2497d03776d4cdf9665cd4f19926c31@172.17.0.1
Mysql/JDBC/PreparedStatement/execute
Exit
mysql-connector-java
localhost:3307
false
sql
cyborg-database-shadow
INSERT INTO t_order (user_id, sku) VALUES (?, ?)/ * cyborg-flow:true */ 
```

压测流量的调用 SQL 详情

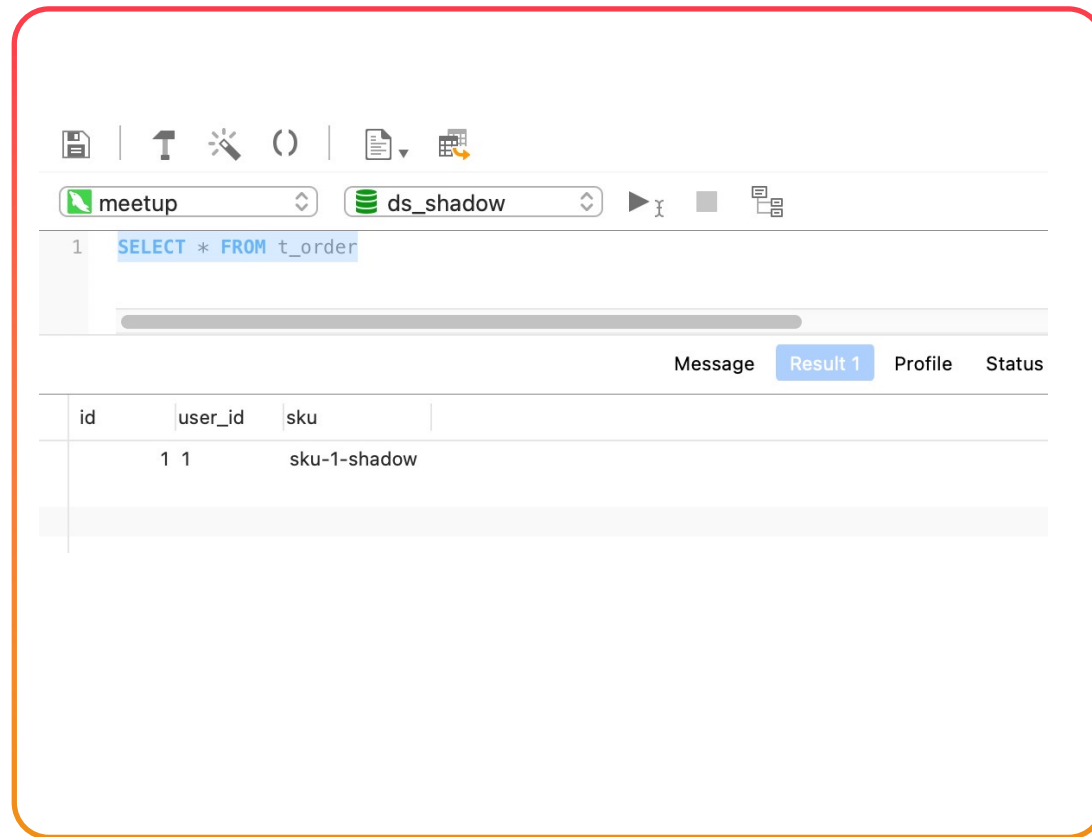
CyborgFlow POC4



The screenshot shows the CyborgFlow interface with the database dropdown set to 'ds'. The SQL query 'SELECT * FROM t_order' is entered. The result table shows one row with columns id, user_id, and sku.

id	user_id	sku
1	1	sku-1-pro

连接 MySQL 生产库 ds 查询



The screenshot shows the CyborgFlow interface with the database dropdown set to 'ds_shadow'. The SQL query 'SELECT * FROM t_order' is entered. The result table shows one row with columns id, user_id, and sku.

id	user_id	sku
1	1	sku-1-shadow

连接 MySQL 影子库 ds_shadow 查询

谢谢观看

2022 年 1 月 15 日



加入交流群



技术干货

