

# 数据库中间件选型测试

爱可生开源社区



# 个人介绍

## 赵红杰

### 开源分布式中间件DBLE测试负责人

## 联系方式

QQ: 47226367

微信: janey001\_zhao

## 目录 CONTENTS

- 一. 数据库中间件选型现状
- 二. 科学的数据库中间件测试
- 三. 中间件的SQL覆盖测试
- 四. 中间件的稳定性测试
- 五. 其它测试点的落地

# 数据库中间件选型测试

已经做过  
进行中  
暂无需求

如果我们还是DBA。。。。

# 如何做数据库中间件选型测试？

**性能测试：** Sysbench /TPCC-MySQL

**功能测试：** 凭经验

**场景测试：** 自家业务压

**极端场景测试：** 拔电源

## 目录 CONTENTS

- 一. 数据库中间件选型现状
- 二. 科学的数据库中间件测试
- 三. 中间件的SQL覆盖测试
- 四. 中间件的稳定性测试
- 五. 其它测试点的落地

# 科学的数据库中间件测试

测试  
需求



测试  
点



测试  
落地

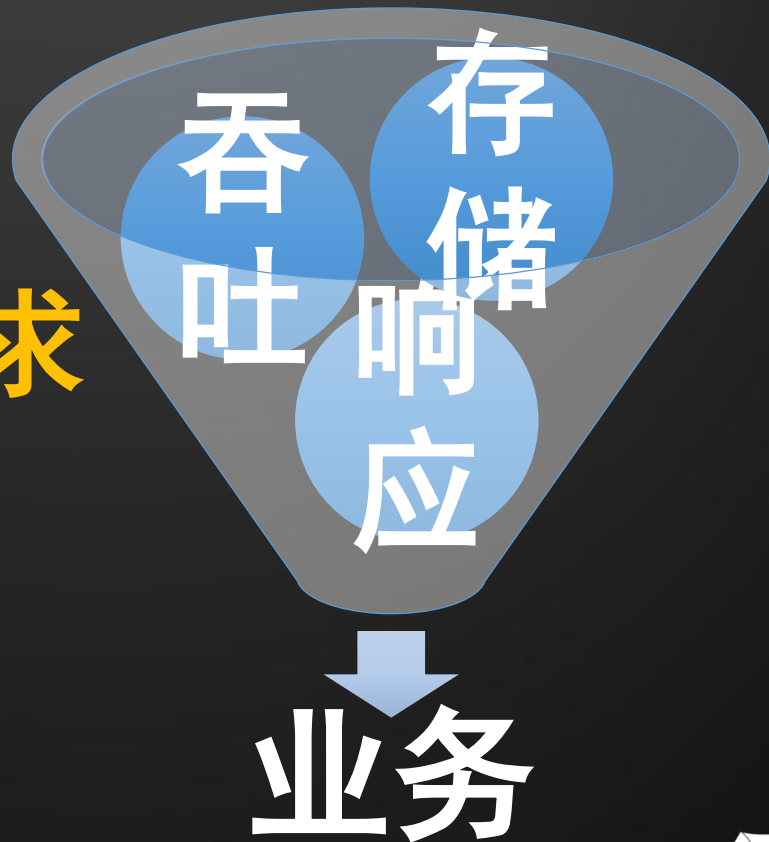
# 科学的数据库中间件测试

测试需求 { 待满足的需求  
现有核心需求  
间接需求



# 科学的数据库中间件测试

待满足的需求



# 科学的数据库中间件测试

## 现有核心需求：SQL支持度

[https://dev.mysql.com/doc/dev/mysql-server/latest/PAGE\\_PREFACE.html](https://dev.mysql.com/doc/dev/mysql-server/latest/PAGE_PREFACE.html)

### Preface

MySQL distributions include a set of test cases and programs for running them.

These tools constitute the MySQL test framework that provides a means for verifying that MySQL Server and its client programs operate according to expectations. The test cases consist mostly of SQL statements. but can also use test language constructs that control how to run tests and verify their results.

This manual describes the MySQL test framework. It describes the programs used to run tests and the language used to write test cases.

# 科学的数据库中间件测试

**间接需求：** 选型的有效性

$$\text{后效} - \text{前效} - \text{成本} > 0$$

# 科学的数据库中间件测试

$$\text{后效} - \text{前效} - \text{成本} > 0$$



# 科学的数据库中间件测试

$$\text{后效} - \text{前效} - \text{成本} > 0$$

兼容性

资源  
复用

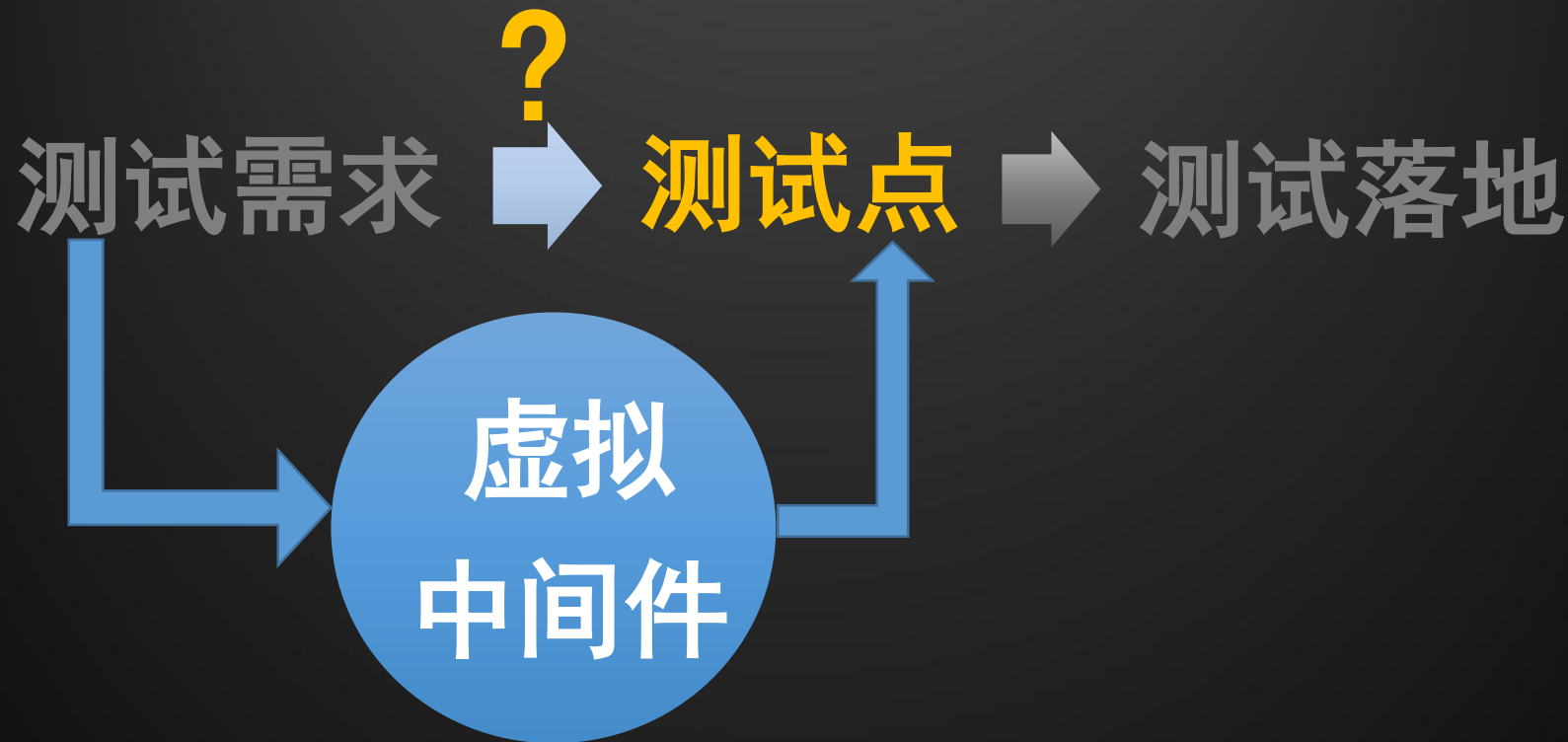
预算

# 科学的数据库中间件测试

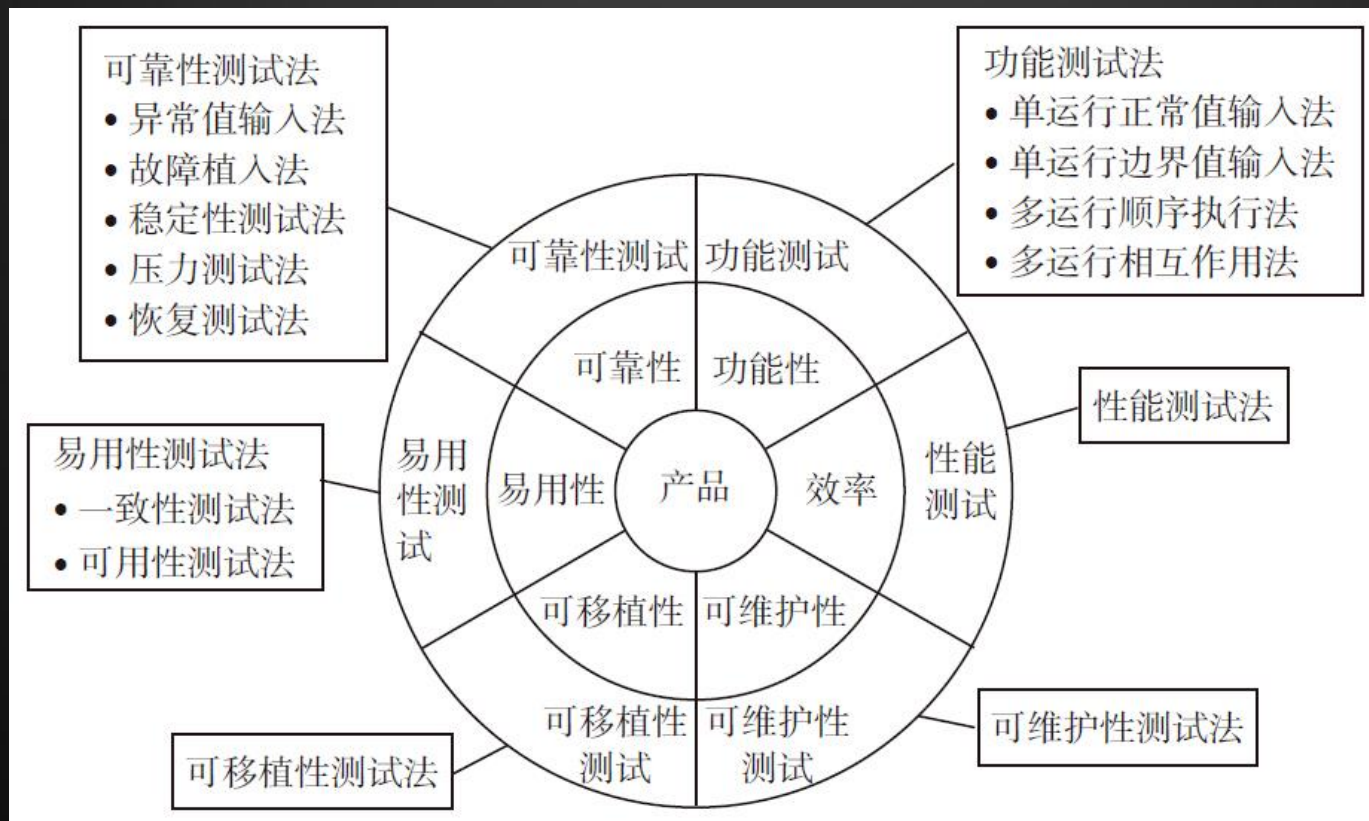
现有核心需求  
待满足的需求 } 测试点

间接需求 } 过滤

# 科学的数据库中间件测试



# 科学的数据库中间件测试





# 科学的数据库中间件测试

- 功能性
  - SQL 覆盖度
  - 负载均衡
  - 分库分表



- 效率—性能测试

# 科学的数据库中间件测试

- 可维护性

运维命令

日志诊断

监控、告警

高可用性

- 可移植性

# 科学的数据库中间件测试

- 易用性

- 可靠性

降级服务

恢复能力

多故障应对

# 科学数据库中间件测试



# 科学的数据库中间件测试



## 目录 CONTENTS

- 一. 数据库中间件选型现状
- 二. 科学的数据库中间件测试
- 三. 中间件的SQL覆盖测试
- 四. 中间件的稳定性测试
- 五. 其它测试点的落地

中间件的SQL覆盖测试

数据库中间件开源测试工具

db|e-test-suite



# 中间件的SQL覆盖测试

## db|e-test-suite 介绍:

- Python 连接支持
- C++ 连接支持
- Java 连接支持&&API测试
- .Net 连接支持



# 中间件的SQL覆盖测试

## db|e-test-suite 介绍:

- C\_MySQL\_API 测试
- SQL 覆盖

# 中间件的SQL覆盖测试

## db le-test-suite 介绍:

- DBLE的功能测试

- SQL覆盖

- 分库分表

- 负载均衡

- 配置

- 大小写敏感性

- 运维命令

# 中间件的SQL覆盖测试

## db le-test-suite 介绍:

- DBLE的功能测试

- 多租户权限
- IP白名单
- 连接池
- SQL黑名单
- Hint功能
- 稳定性

# 中间件的SQL覆盖测试

## db|e-test-suite中的SQL覆盖

- 根据MySQL **官方**文档整理而成
- **50+** SQL覆盖文件
- SQL文件可**自由组合**
- **1h+** 输出测试结果
- 收集线上bug, **动态补充**SQL集

# 中间件的SQL覆盖测试

## 根据MySQL官方文档整理而成

- Data Types
- Character Sets
- Functions and Operators
- SQL Statement Syntax

# 中间件的SQL覆盖测试

## 50+ SQL覆盖文件, 9000+语句

data\_types\_1.sql

data\_types\_2.sql

delete.sql

identifiers\_util.sql

insert.sql

insert\_on\_duplicate\_key.sql

insert\_value.sql

loaddata.sql

prepare.sql

replace.sql

reserved\_words.sql

set\_names\_character.sql

set\_server\_var\_util.sql

set\_user\_var\_util.sql

show.sql

show\_dble.sql

sys\_function\_util.sql

truncate.sql

```
5 CREATE TABLE schema2.test2( id int(10) unsigned NOT NULL,"k" int(10) unsigned NOT NULL)
6 #*****lock in share mode***** s lock*****
7 #!sql_thread_1
8 set @@session.innodb_lock_wait_timeout=10000;
9 SET @@session.autocommit = ON;
10 delete from test1;
11 insert into test1 values(1,1,1,1),(2,2,2,2),(3,3,3,3),(4,4,4,4),(5,5,5,5),(6,6,6,6);
12 begin;
13 select * from test1 order by id lock in share mode;
14 #!sql_thread_2
15 set @@session.innodb_lock_wait_timeout=10000;
16 SET @@session.autocommit = ON;
17 begin;
18 select * from test1 order by id lock in share mode;
19 #!sql_thread_1
20 commit;
21 #!sql_thread_2
22 select * from test1 order by id;
23 commit;
24 select * from test1 order by id;
25 #*****
26 #!sql_thread_1
```

# 中间件的SQL覆盖测试

## SQL文件可自由组合

```
Scenario Outline:sql cover for mixed table types #2
  Given set sql cover log dir "sql_cover_mixed"
  Then execute sql in file "<filename>"
  Given clear dirty data yield by sql
```

### Examples:Types

```
| filename
| sqls_mixed/select/join_different_rules_shardings.sql
| sqls_mixed/select/join_global_nosharding.sql
| sqls_mixed/select/join_global_sharding.sql
| sqls_mixed/select/join_global_sharding_nosharding.sql
| sqls_mixed/select/join_no_er.sql
| sqls_mixed/select/join_sharding_nosharding.sql
| sqls_mixed/select/join_shardings.sql
| sqls_mixed/select/subquery.sql
| sqls_mixed/select/subquery_dev.sql
| sqls_mixed/select/subquery_global_noshard.sql
| sqls_mixed/select/subquery_no_er.sql
| sqls_mixed/select/subquery_shard_global.sql
| sqls_mixed/select/subquery_shard_noshard.sql
| sqls_mixed/syntax/character.sql
| sqls_mixed/syntax/create_index.sql
| sqls_mixed/syntax/identifiers.sql
```



# 中间件的SQL覆盖测试

## 1h+ 输出测试结果

Job Details | autotest-dble-master | 784 VSM | sql\_support / 1 | Sqls

---

SCHEDULED ON: 22 May 2019 at 13:34:57 Local Time AGENT: autotest-dble-master

COMPLETED ON: 22 May 2019 at 14:51:54 Local Time [more...](#) BUILD: 1

DURATION: 1h 16m 35.0s

---

Console Tests Failures Artifacts Materials Properties

---

Change theme [Full Screen](#)

```
▶ [go] Job Started: 2019-05-22 05:35:17 UTC
▶ [go] Start to prepare autotest-dble-master/784/sql_support/1/sqls on 1
▶ [go] Start to build autotest-dble-master/784/sql_support/1/sqls on 1
✓▶ [go] Task: bash resetReplication.sh took: 2m 8.934s
▶ [go] Task: bash resetReplication.sh took: 2m 8.934s
```



# 中间件的SQL覆盖测试

MySQL	中间件	结果是否一致	生成的日志	关注度
✓	✓	✓	xx.pass.log	
✗	✗	✓	xx.warn.log	
✓	✓	✗	xx.fail.log	  
✗	✗	✗	xx.serious_warn.log	 
✗	✓	✗	xx.fail.log	  
✓	✗	✗	xx.fail.log	  
✗	✓	✓	不可能	
✓	✗	✓	不可能	

# 测试结果分类 - xxx.fail.log

# 中间件的SQL覆盖测试

测试环境一键部署

```
bash start_env.sh
```

## 目录 CONTENTS

- 一. 数据库中间件选型现状
- 二. 科学的数据库中间件测试
- 三. 中间件的SQL覆盖测试
- 四. 中间件的稳定性测试
- 五. 其它测试点的落地

# 中间件的稳定性测试

**稳定性：** 应对风险的能力

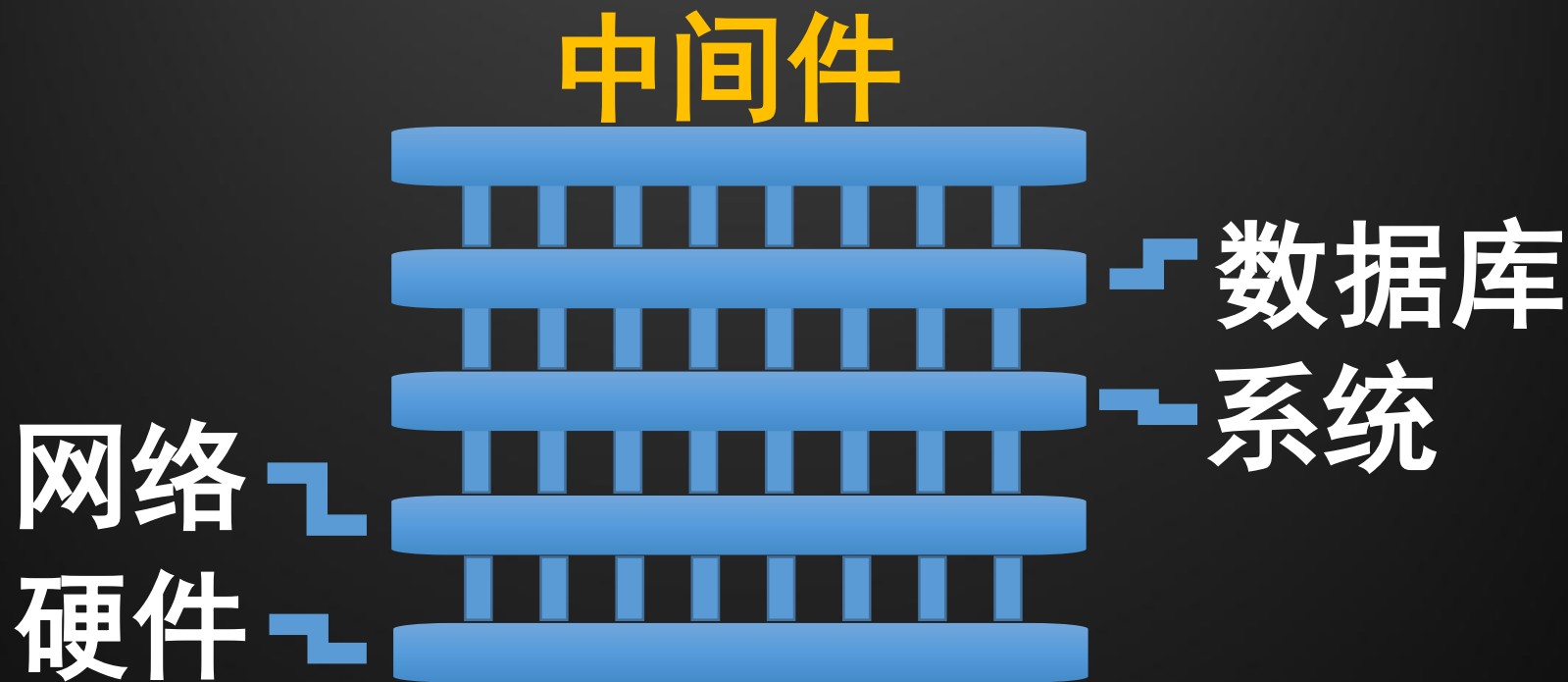
多样、奇葩、极端场景应对能力  
怎么测？

# 中间件的稳定性测试

## Chaos engineering

混沌工程是一门相对**高级**的系统稳定性治理方法论，它提倡采用**探索式**的研究实验，发现**生产环境**中的各种**风险**。

# 中间件的稳定性测试



# 中间件的稳定性测试

## 2019QCON-Beijing的混沌测试分享

- 亚马逊:AWS 云上混沌工程实践之对照实验设计和实施
- 阿里:云原生架构下的混沌工程实践
- 腾讯:百万级服务器反入侵场景的混沌工程实践



# 中间件的稳定性测试

## 共同点:

1. 控制爆炸半径
2. 降级服务的能力
3. 避免级联故障，非核心业务拖垮核心业务
4. 故障场景按层分类
5. 借助框架、工具实现混沌测试平台化

# 中间件的稳定性测试

## DBLE的混沌计划:

- DDL+CPU乱序
- 大数据包+内存/网络/I/O
- 业务+运维命令
- 分布式DDL+CPU乱序
- 分布式事务+MySQL异常
- ...

## 目录 CONTENTS

- 一. 数据库中间件选型现状
- 二. 科学的数据库中间件测试
- 三. 中间件的SQL覆盖测试
- 四. 中间件的稳定性测试
- 五. 其它测试点的落地

# 其它测试点的落地



数据传输组件DTLE

分布式组件DBLE

全局事务框架TXLE

往期分享Slides

MySQL 8.0.16 组复制通讯

MySQL 8.0.16 新增的两个 UDF

MySQL 8.0.16 组复制通讯一起阅读。

开源项目

最新资讯

技术文章

# ■ 关于爱可生开源社区

**Dble-test-suite 项目地址:**

**[github.com/actiontech/dble-test-suite](https://github.com/actiontech/dble-test-suite)**

**开源QQ群号: 669663113 (DBLE官方社区交流群)**

# 关于爱可生开源社区



## 微信公众号「爱可生开源社区」

- 每工作日1篇**MySQL**前沿技术文章
- 每月发布**开源组件新版**
- 每年**1024开源**一款企业级组件
- 2019年规划**25场社区活动**