

# KunlunBase读写分离方案

何革新

泽拓科技（深圳）有限责任公司

# 目录

COMPANY

01

为什么要读写分离

02

常见读写分离方案

03

KunlunBase 读写分离

04

演示

读写分离的好处：

- 1.通过将“读”操作和“写”操作分离到不同的服务器上，降低对主服务器的CPU、内存、存储、网络资源的争用；
- 2.主服务器的增删改进行时，不影响查询服务器的查询，降低阻塞的发生，提高了并发性；
- 3.在应用程序提交了报表请求、不合理的查询请求时，不会导致长时间的锁表；
- 4.提高整个系统的投资回报率；

## 常见读写分离方案

1.数据访问层实现：初始化两个ORM操作，一个做读，另外一个做写，然后依据业务对数据库操作属性调用相应的ORM

缺点：需要修改业务代码，写完读”不一致问题，server方法，先向主库insert语句，然后立刻select，却查不出数据；

2.JDBC 层实现：JDBC层的接口函数进行重写，会有一个对业务层暴露的JDBCProxy，它通过读写决策器进行选择此时是使用读还是写连接，JDBCWriter以及JDBCReader都是对JDBC接口的一个实现。

缺点：需要配置到业务代码中去，且主备切换后需要修改配置

优点：SQL 解析和事务管理，自动实现读写分离，解决”写完读”不一致的问题

3. Proxy 代理层：ProxySQL

缺点：需要额外组件

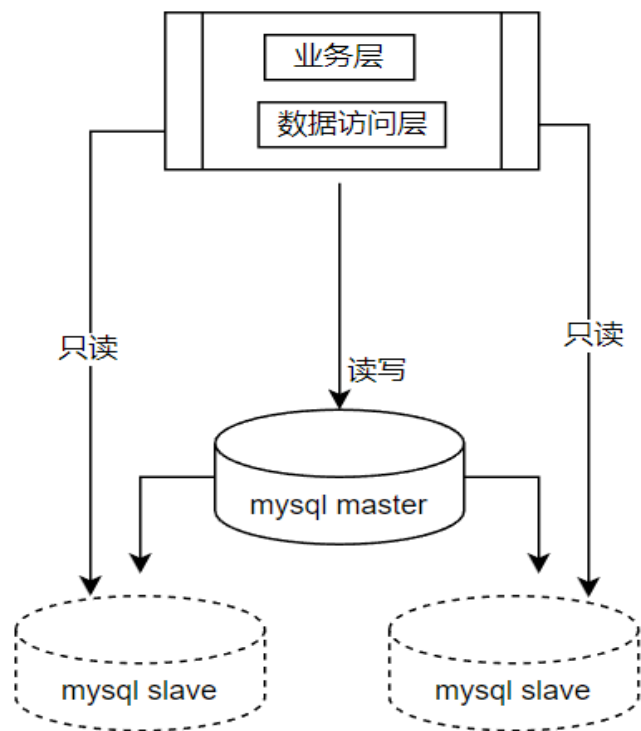
优点：对业务无侵入

4. 数据库层智能分离：由数据库根据配置策略做读写分离

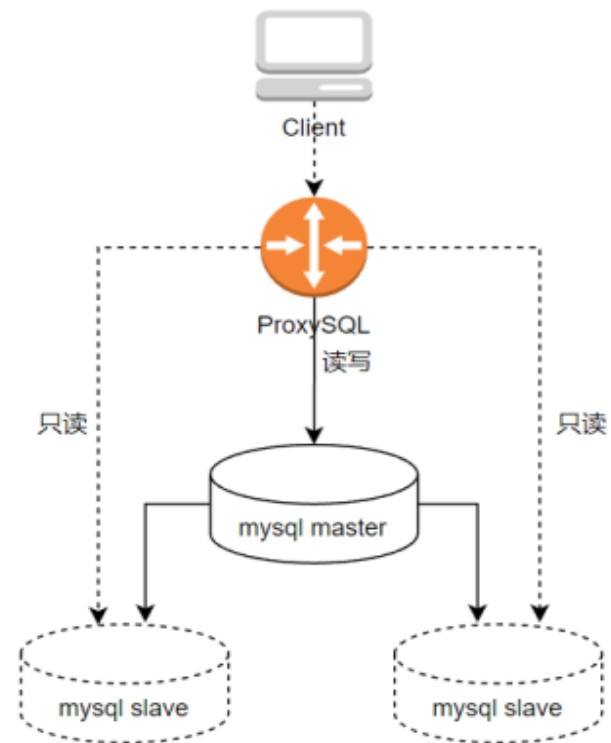
优点：对业务无侵入，自动实现读写分离，支持事务内一致性，解决”写完读”不一致的问题，自动识别主备机

限制：主备延迟，不同事务读到的数据可能不一致

- 1) 需要修改业务代码
- 2) 不能感知主从切换
- 3) 写完读一致性问题
- 4) 需要部署一个中间件，规则配置在中间件
- 5) 模拟一个 MySQL 服务器，对业务系统无侵入



业务层



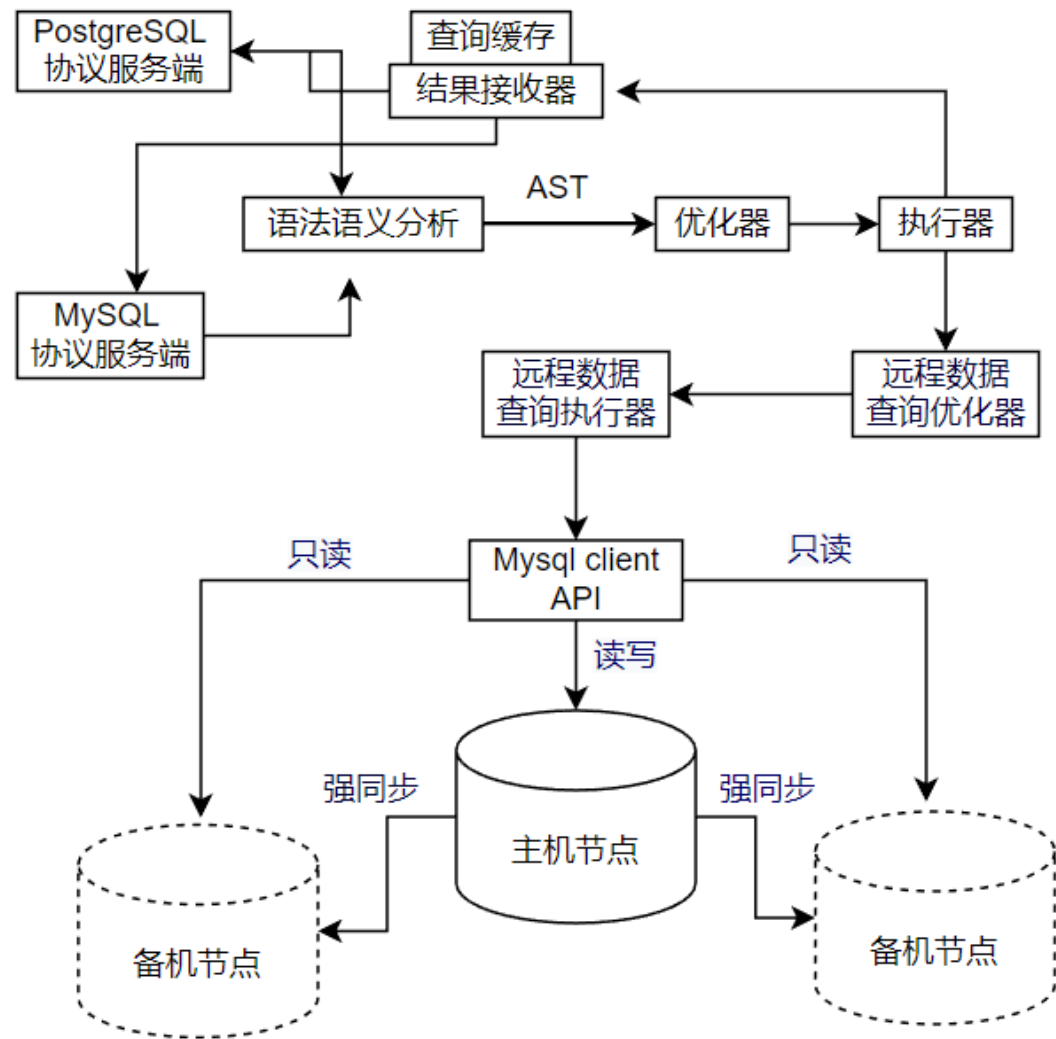
代理层

## KunlunBase 读写分离方案

启用：enable\_replica\_read = on

条件：

(1) 当前SQL类型为select；(2) SQL中不包含用户自定义函数（即create function语句创建的函数），除非当前事务为只读事务；(3) 如果不在事务中（autocommit=on），则允许读写分离；如果语句在显示事务中，则要满足：（3.1）如果在只读事务中，则允许读写分离；（3.2）如果在读写事务中，则该事务尚未更新过任何数据；远程查询优化器就会将相应的SQL 执行计划下发到从备机的节点上执行



## KunlunBase 读写分离方案

- 1) 根据节点权重值选择 ( ro\_weight)
- 2) 根据网络延迟(ping)
- 3) 根据主从副本的数据一致性延迟 ( latency)

配置首先 shard1 ,节点3 作为备机

```
postgres=# update pg_shard_node set ro_weight=2 where port=6006;
UPDATE 1
postgres=# select * from pg_shard_node ;
 id | port | shard_id | svr_node_id | ro_weight | ping | latency | user_name | hostaddr |
-----+-----+-----+-----+-----+-----+-----+-----+-----+
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
 1 | 6004 |      1 |          0 |          1 | 0 |      0 | pgx      | 127.0.0.1 |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
 2 | 6005 |      1 |          0 |          1 | 0 |      0 | pgx      | 127.0.0.1 |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
 4 | 6007 |      2 |          0 |          1 | 0 |      0 | pgx      | 127.0.0.1 |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
 5 | 6008 |      2 |          0 |          1 | 0 |      0 | pgx      | 127.0.0.1 |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
 6 | 6009 |      2 |          0 |          1 | 0 |      0 | pgx      | 127.0.0.1 |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
 3 | 6006 |      1 |          0 |          2 | 0 |      0 | pgx      | 127.0.0.1 |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
(6 rows)
```

**replica\_read\_ping\_threshold**, 计算节点到备节点的ping延迟阈值, **0**表示不关心;

**replica\_read\_latency\_threshold**, 主备同步延迟的阈值, **0**表示不关系;

**replica\_read\_order**, 选择备机优先级: **0**, 按权重; **1**、按ping延迟; **2**、按主备同步

延迟; **replica\_read\_fallback**, 备机选择的回退策略(如果备机不能访问);

**replica\_read\_fallback=0**, 直接报错, **replica\_read\_fallback=1**, 任意选择一个备机进行访问,  
**replica\_read\_fallback=2**, 选择主机进行访问。



## KunlunBase 读写分离方案1

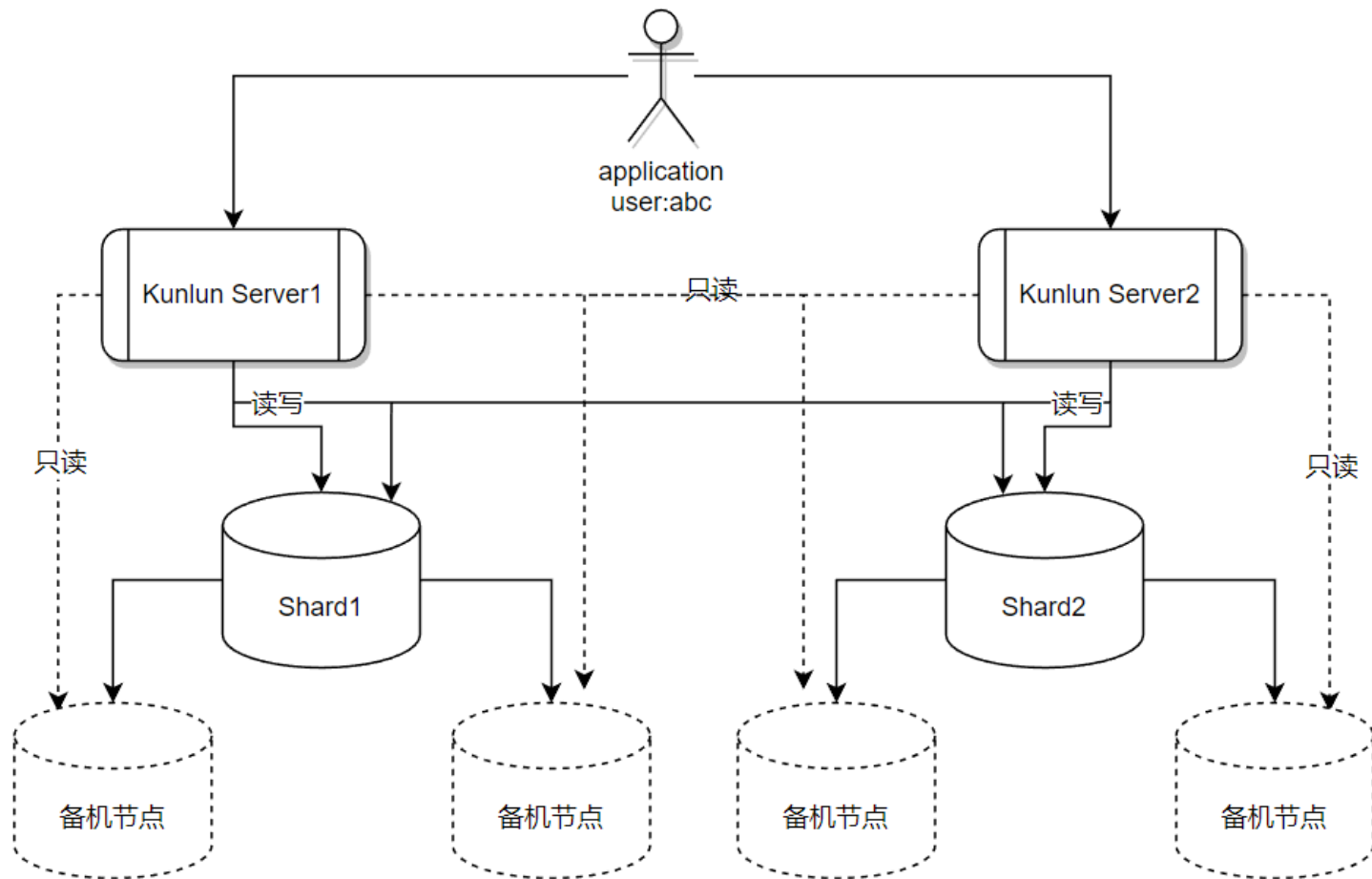
场景说明：某OLTP业务系统有大量的查询操作，在业务高峰期数据库响应速度变慢，导致了业务的性能问题。经检查发现存储节点主副本的存储节点的IO资源利用到达瓶颈，但备机的IO资源利用率低。

环境说明：集群环境2个计算节点，2个shard，每个shard是一主两从的高可用架构

配置方案：将该应用的select（非事务中的及纯只读事务中的）操作路由到备机执行。

实现方案：设置用户属性：

```
alter user abc set enable_replica_read = on;
```



## KunlunBase 读写分离方案2

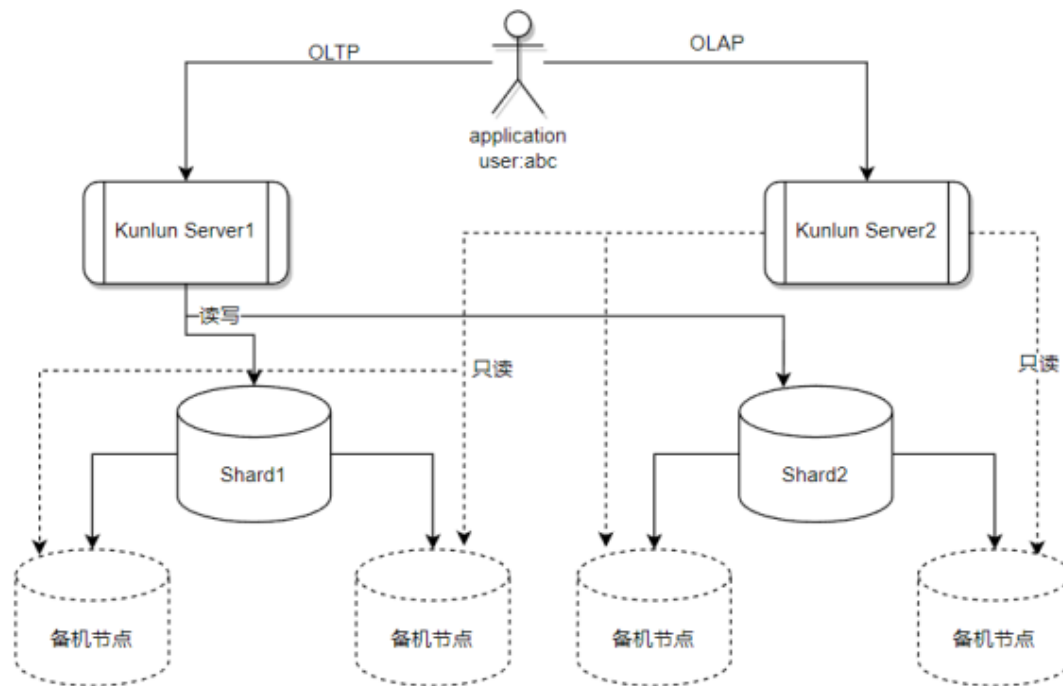
场景说明: **OLTP & OLAP** 业务分离

环境说明: 集群环境**2**个计算节点, **1**个计算节点执行**OLTP**, **1**个计算节点执行**OLAP**

配置方案: 为两个业务配置不同的两个用户

实现方案: 设置OLAP 用户属性:

```
alter user abc set enable_replica_read = on;
```



登录集群：docker 环境

查看是否启动读写分离

设置读写分离

执行语句：select \* from t1;

查看计算节点日志：/kunlun/server\_datadir/instance\_data/comp\_datadir/5401/log

关闭读写分离

执行语句

查看日志

设置权重

执行语句，查看日志

关闭备机，

执行语句

查看日志

# Thank you

## Q & A

