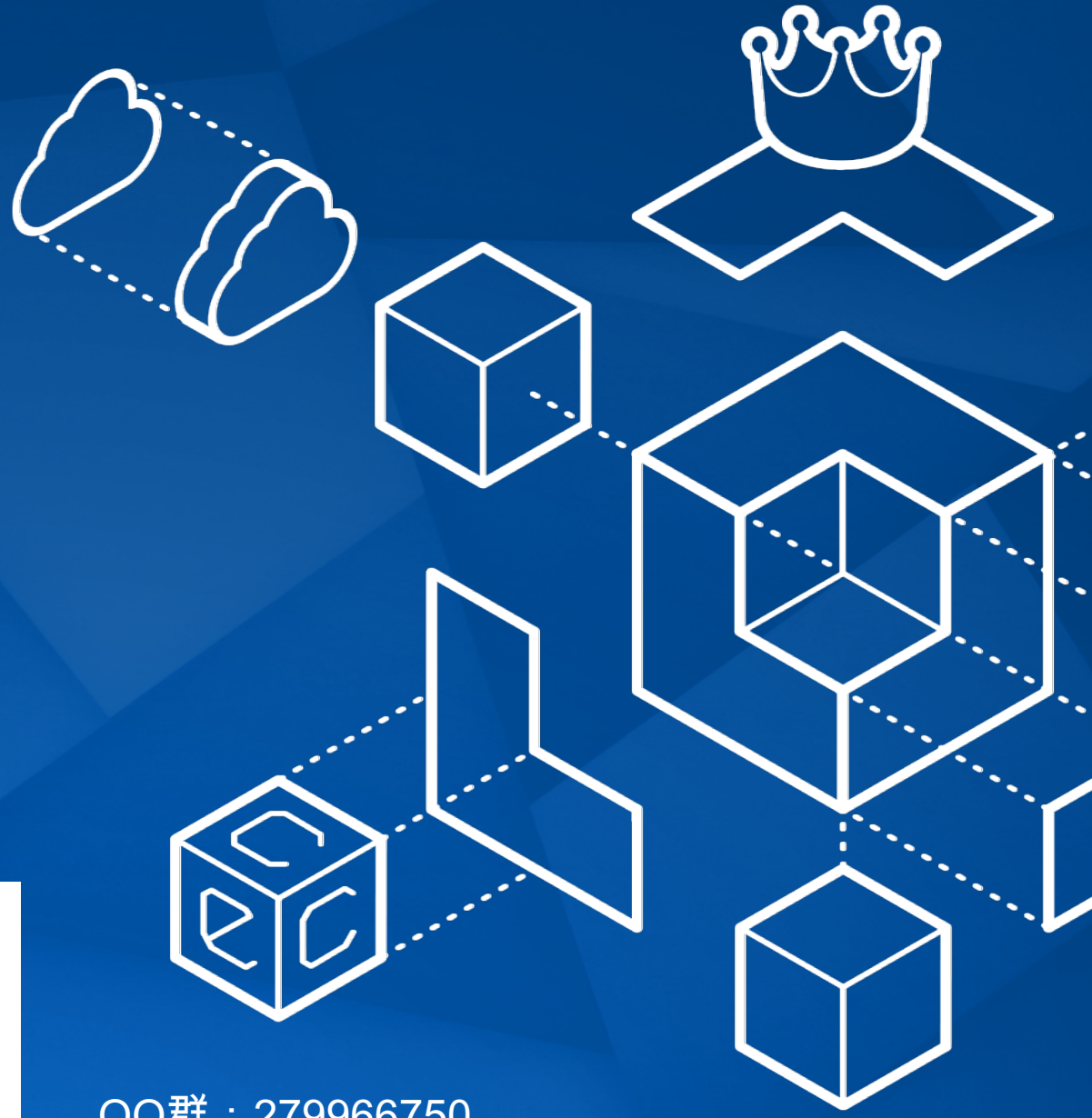
hybris数据库
读写分离和分库分表初探

Winston Zhang
Nov 28, 2015



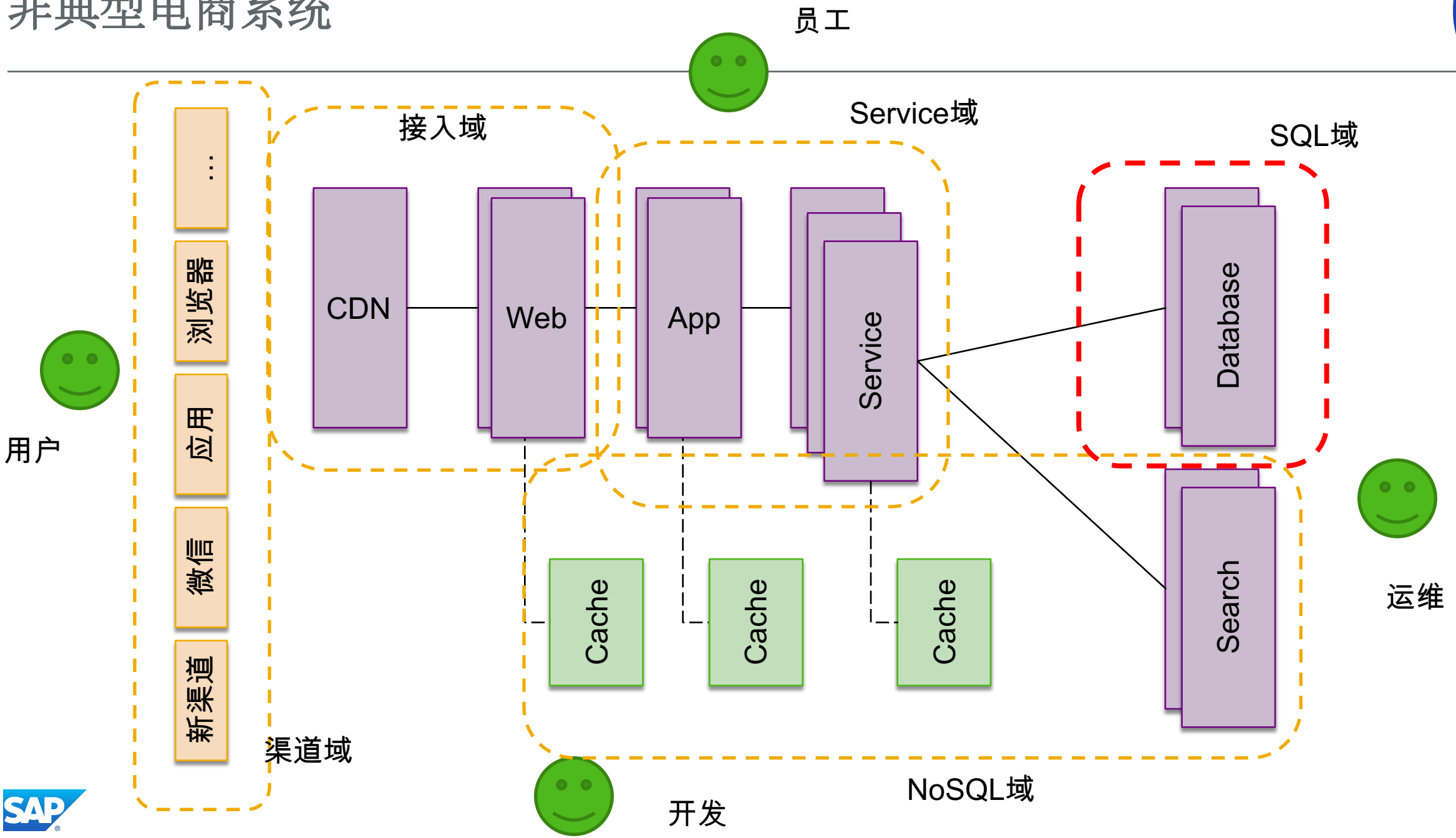
QQ群：279966750

社区：52hybris.com

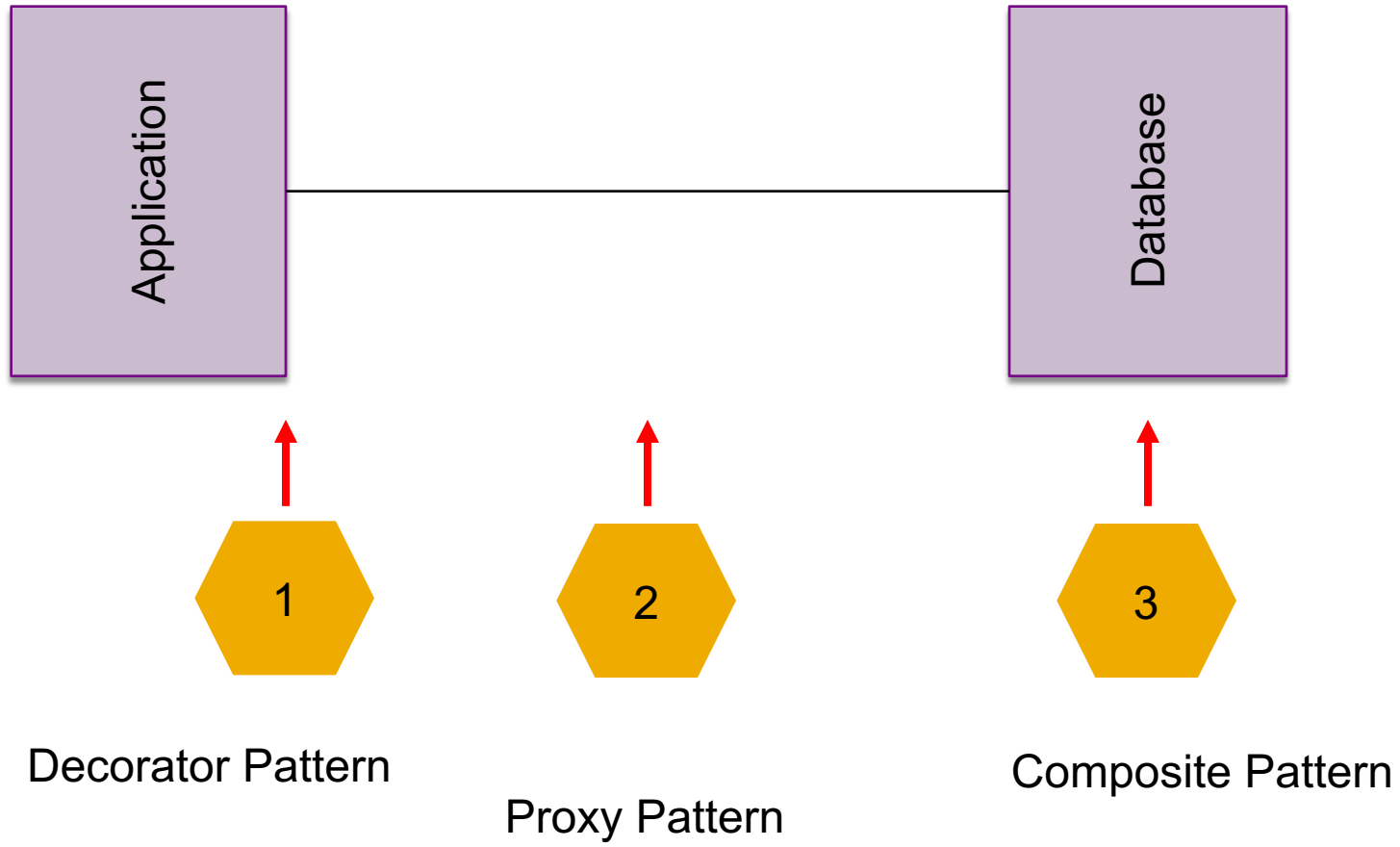
Agenda

1. 从大规模可扩展电商系统说起
2. Database scalability patterns
3. Scale out Read & Write
4. Database Sharding
5. Hybris recommendation

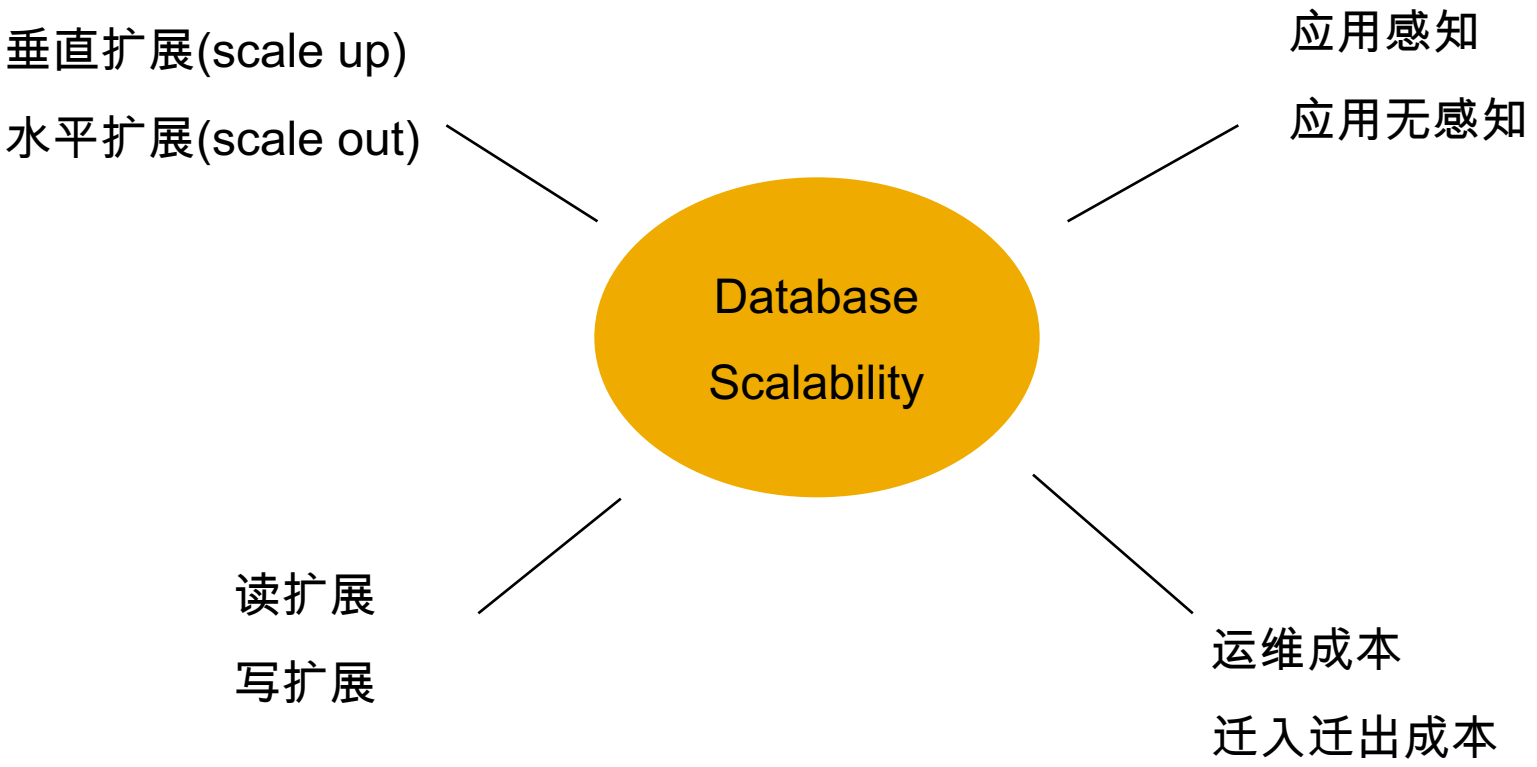
非典型电商系统


员工
接入域
Service域
SQL域
…
浏览器
应用
微信
新渠道
渠道域
CDN
Web
App
Service
Database
Search
Cache
Cache
Cache
用户
运维
开发
NoSQL域

Database Scalability Patterns

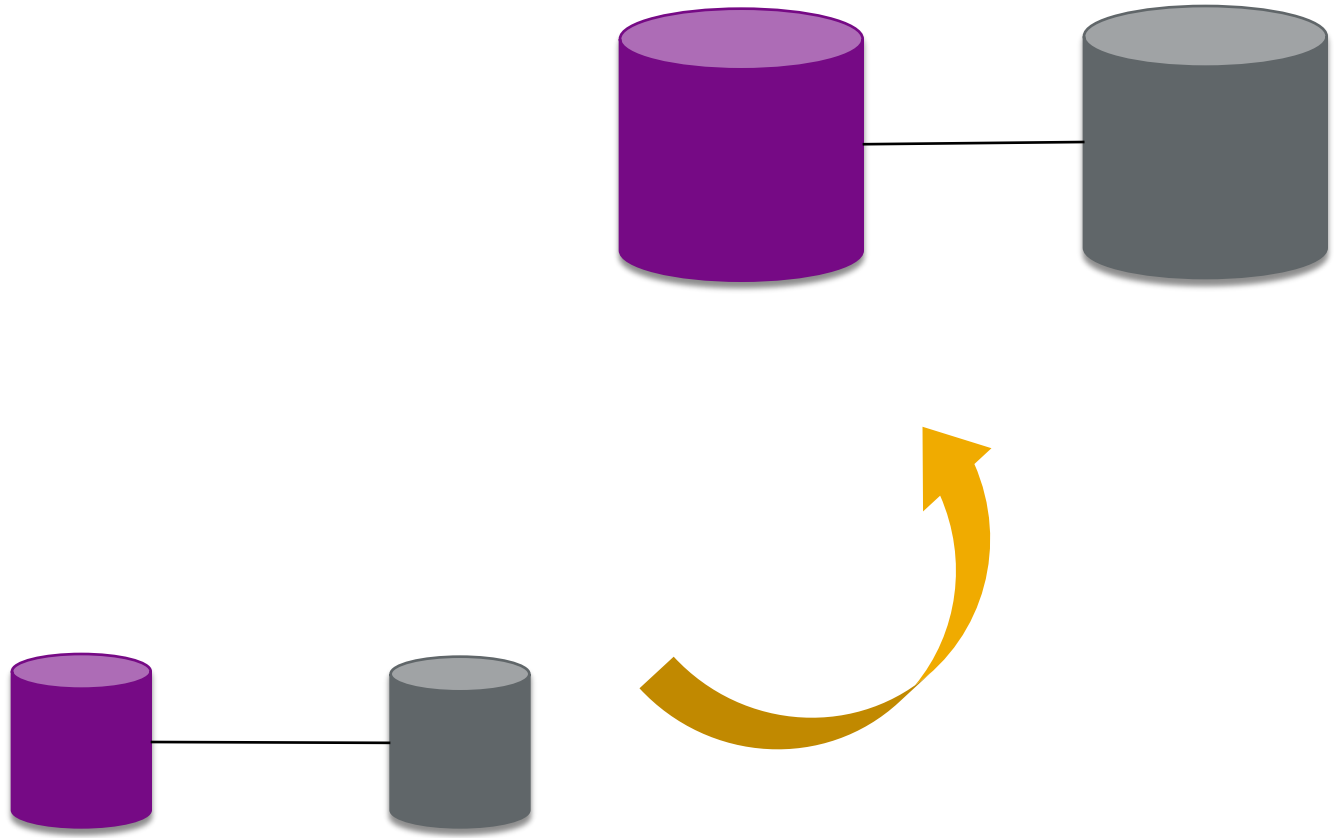


数据库扩展性考量点

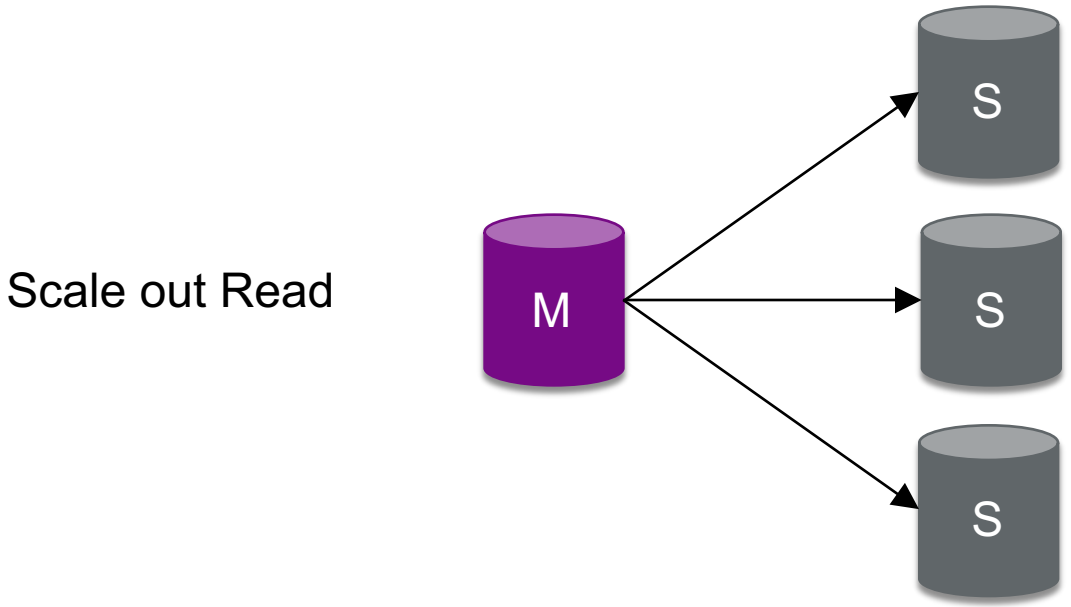

垂直扩展(scale up)
水平扩展(scale out)
应用感知
应用无感知
Database
Scalability
读扩展
写扩展
运维成本
迁入迁出成本

Scale up (Read + Write)

“去IOE”



Scale out Read



Hybris读写分离

[y]

1. JDBC Driver + Code (Driver依赖性强)
2. Code Scatter
3. Code Aspect

How 怎么样读写分离

When 什么时候读写分离

[y]

1. Spring AOP
2. Filter

Hybris datasource configuration

```
db.url=jdbc:mysql://perc56.<youraddress>.com/<dbname>?useConfigs=maxPerformance&cha
racterEncoding=utf8
db.driver=com.mysql.jdbc.Driver
db.username=<username>
db.password=<password>
db.tableprefix=
mysql.optional.tabledefs=CHARSET=utf8 COLLATE=utf8_bin
mysql.tabletype=InnoDB
db.customsessionsql=SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
mysql.allow.fractional.seconds=true
slave.datasource.1.db.url=jdbc:mysql://perc56read.<youraddress>.com/<dbname>?useCon
figs=maxPerformance&characterEncoding=utf8
slave.datasource.1.db.driver=com.mysql.jdbc.Driver
slave.datasource.1.db.username=<username>
slave.datasource.1.db.password=<password>
slave.datasource.1.db.tableprefix=
slave.datasource.1.mysql.optional.tabledefs=CHARSET=utf8 COLLATE=utf8_bin
slave.datasource.1.mysql.tabletype=InnoDB
slave.datasource.1.db.customsessionsql=SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
slave.datasource.1.mysql.allow.fractional.seconds=true
```

DataSource (master/slave)

业务逻辑

```
…
tenant.getDataSource()
…
```

```
@Override
public HybrisDataSource getDataSource()
{
   // this is thread safe because getMasterDataSource() relies on volatile 'state'
   HybrisDataSource ret = getMasterDataSource();
   if (hasAltDataSource())
   {
      final DataSourceSelection currentAlternativeDS = getThreadDataSource();
      if (currentAlternativeDS != null)
      {
         // allow alternative data source only if
         // - it has no connection error
         // - is is master or forceMaster mode is not active
         if (currentAlternativeDS.canUseDataSource())
         {
            ret = currentAlternativeDS.getDataSource();
         }
      }
   }
   return ret;
}
```

Feature list:

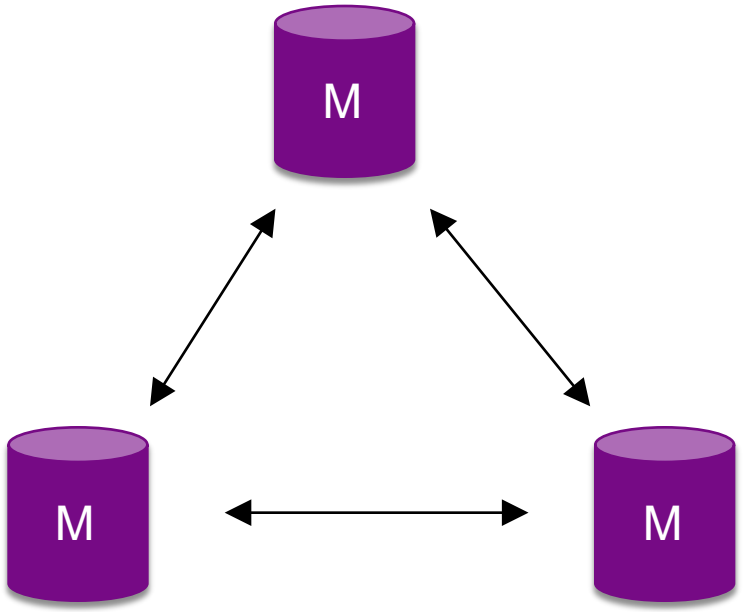
- All transactional db call, go master (can force to slave) 强读走写
- All non-transaction db call, go slave (can force to master) 弱读走读
- Always master (numseries generator) 强制走写
- Always slave 强制走读
- Round robin balance slave
- Delay and retry interval
- If slave down, switch to master automatically
- If master down, fail
- Can specific which tables are splitting (inclusive or exclusive)
- Supports all databases 支持所有hybris支持的数据库(Oracle, MySQL, SQL Server, HANA)

Scale out Write

Percona XtraDB Cluster / MariaDB Cluster

Oracle RAC

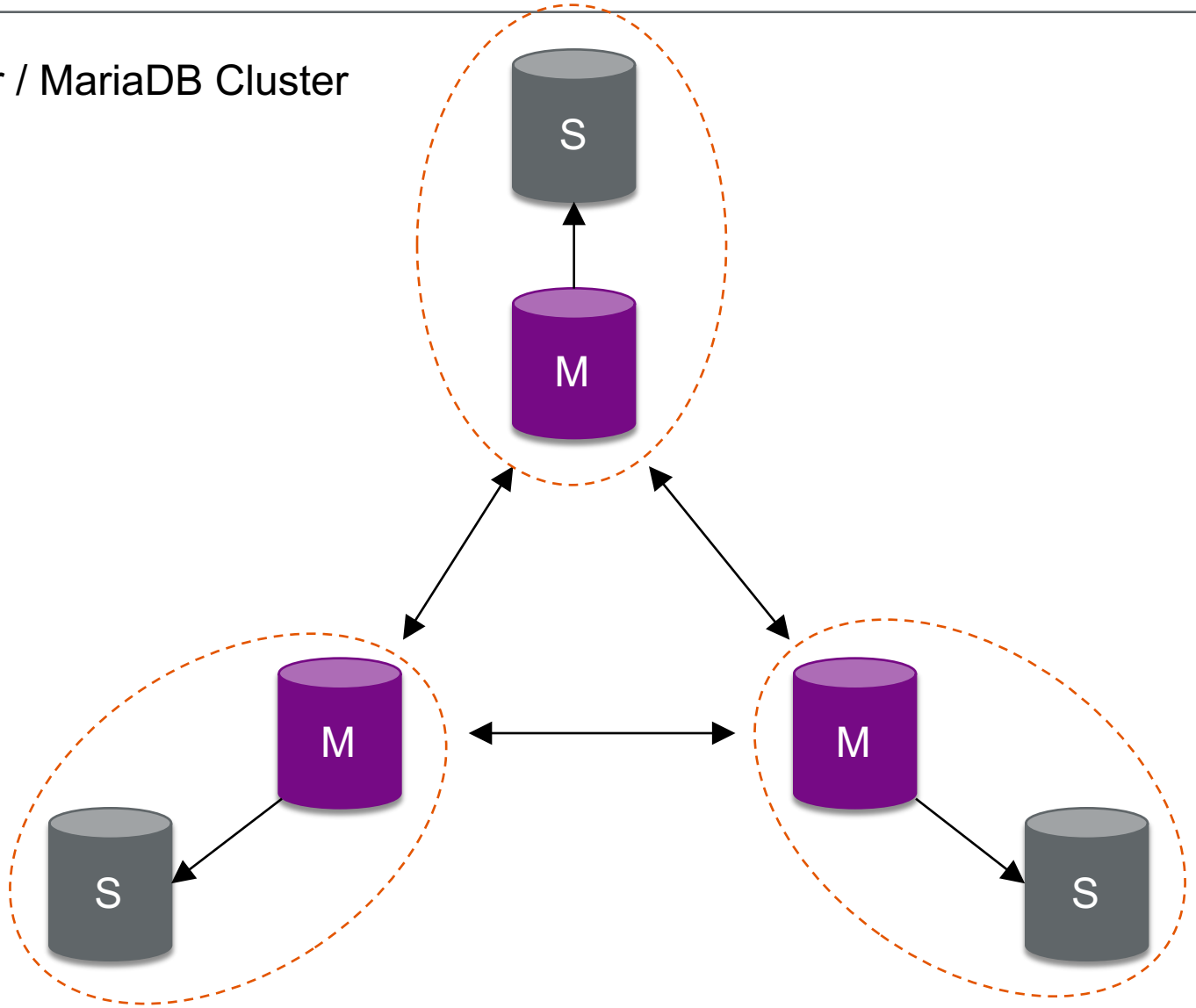
Scale out Write



Scale out Write

Percona XtraDB Cluster / MariaDB Cluster

Scale out Read & Write



Hybris读写分离+多写

❌ db.url=jdbc:mysql:**loadbalance**://node01:3306,node02:3306,node03:3306/hybris?
useConfigs=maxPerformance&
characterEncoding=utf8&
loadBalanceConnectionGroup=first&
loadBalanceEnableJMX=true

✅ db.url=jdbc:mysql://node02:3306,node03:3306,node01:3306/hybris?
useConfigs=maxPerformance&
characterEncoding=utf8&
failOverReadOnly=false&
autoReconnect=true

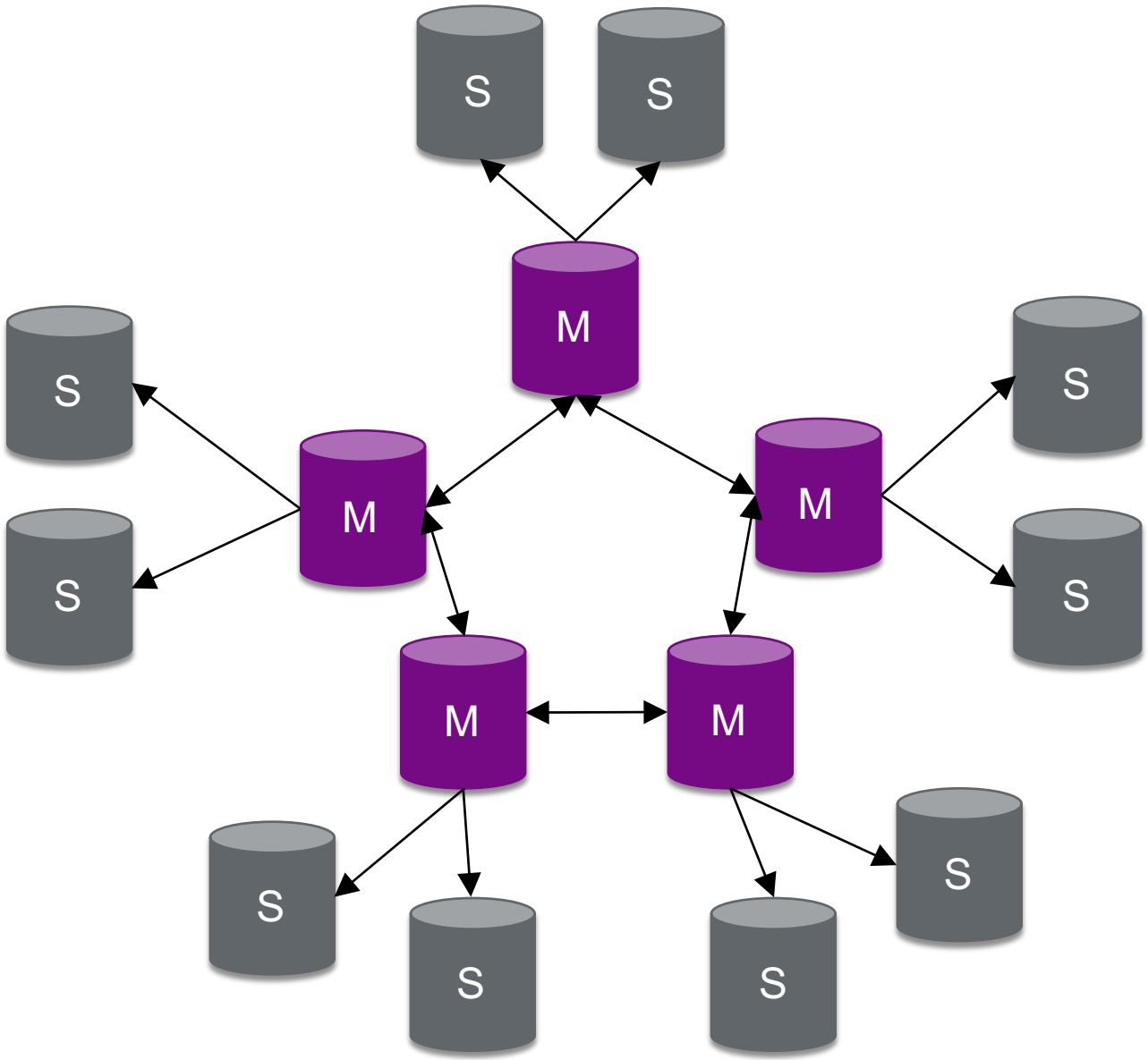
Features	load balance url	failover url
balance in cluster	Yes	Yes (manually configured)
balance in node	Yes	No
failover	Yes	Yes
Init errors	lots of	none
Recommend to use	⛔	✅

JDBC replication url

```
jdbc:mysql:replication://[master host][:port],[slave host
1][:port][,[slave host 2][:port]]...[/[database]] »
[?propertyName1=propertyValue1[&propertyName2=propertyValue2]...]
```

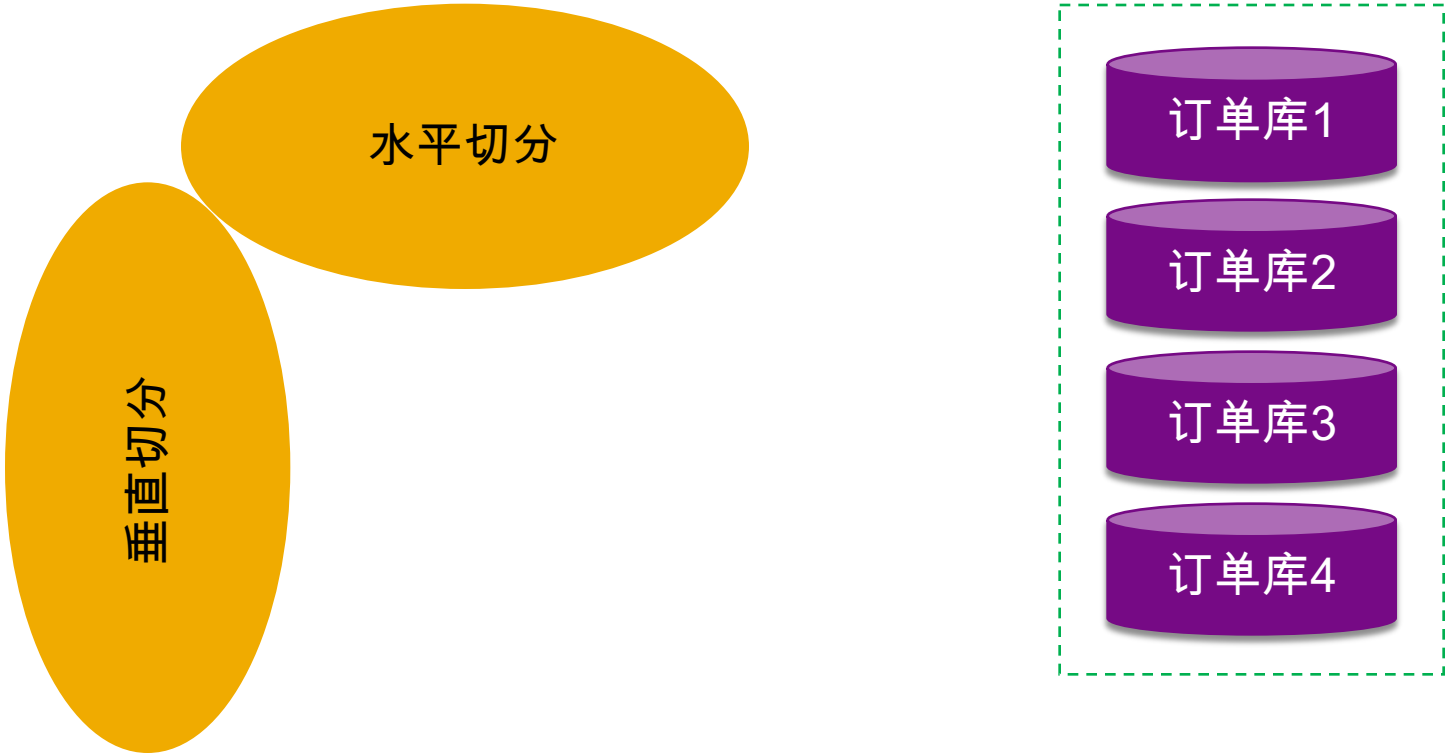
```
jdbc:mysql://
address=(type=master)(host=master1host),
address=(type=master)(host=master2host),
address=(type=slave)(host=slave1host)
/database
```

Is it enough?

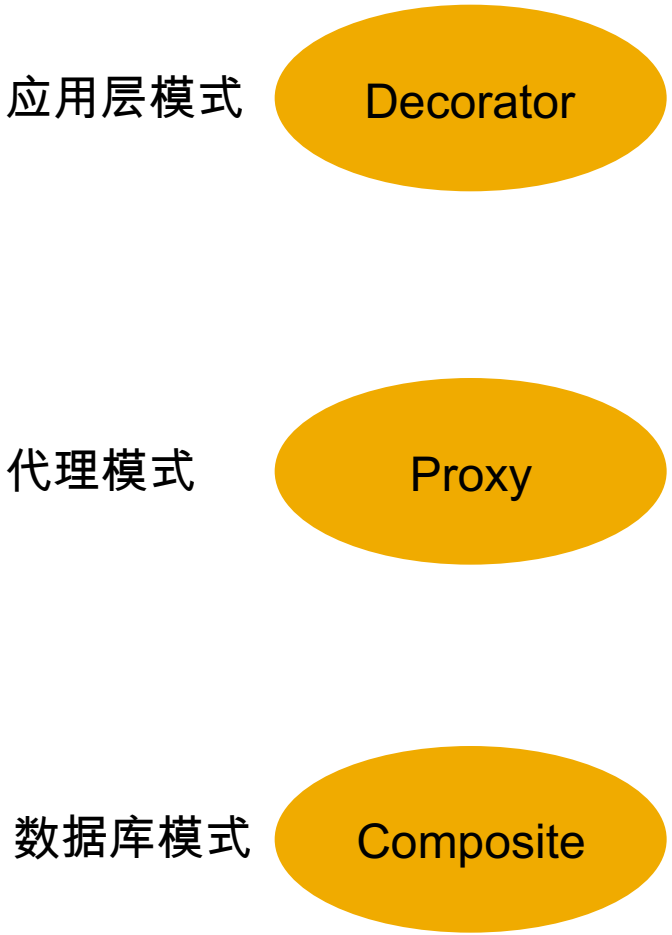


Database sharding

分库分表



Sharding solution alternatives



TDDL

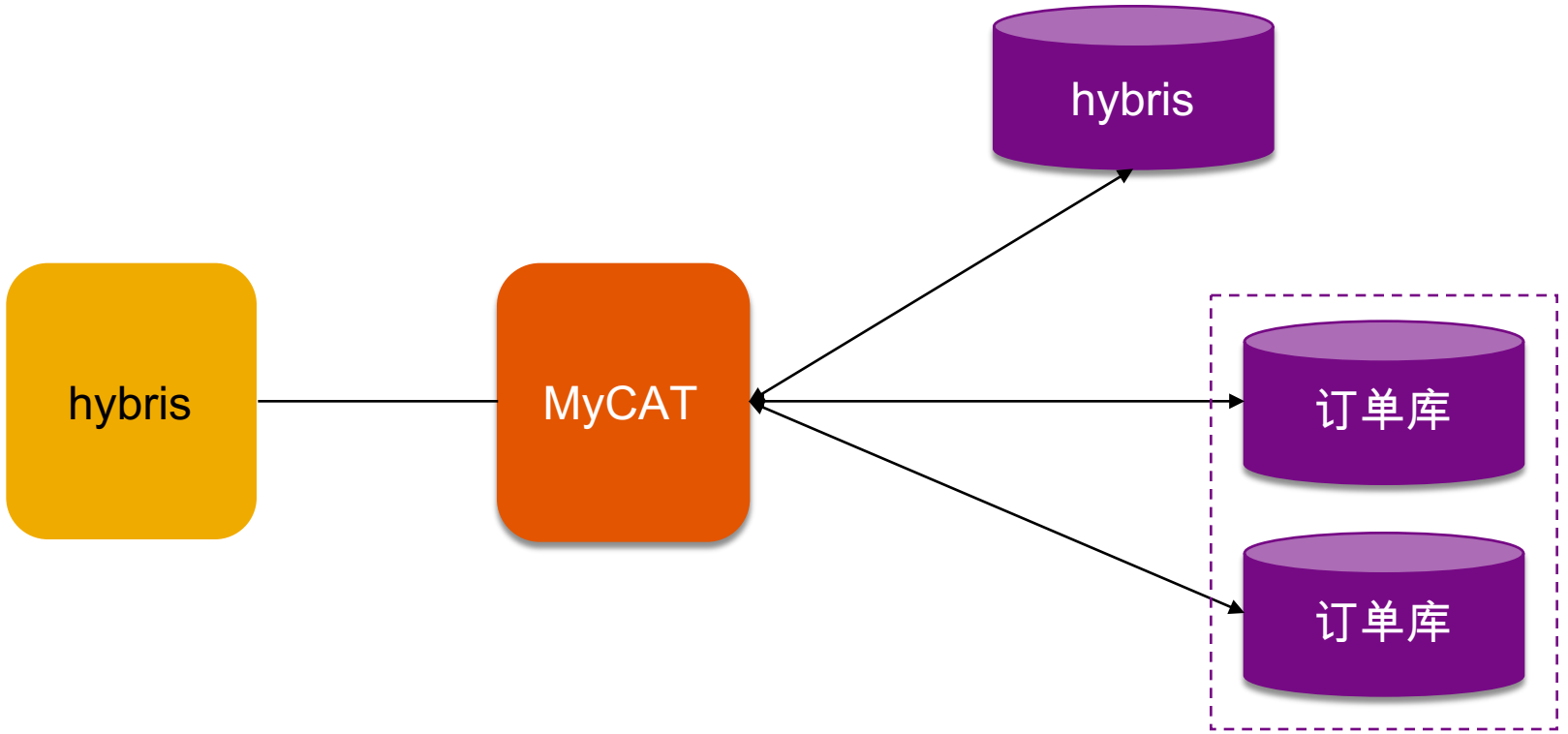
MySQL Fabric

- MySQL Proxy
- Vitess
- MyCat (Amoeba -> Cobar -> MyCat)
- DRDS (Cobar /TDDL -> DRDS)
- Cobar
- KingShard
- OneProxy

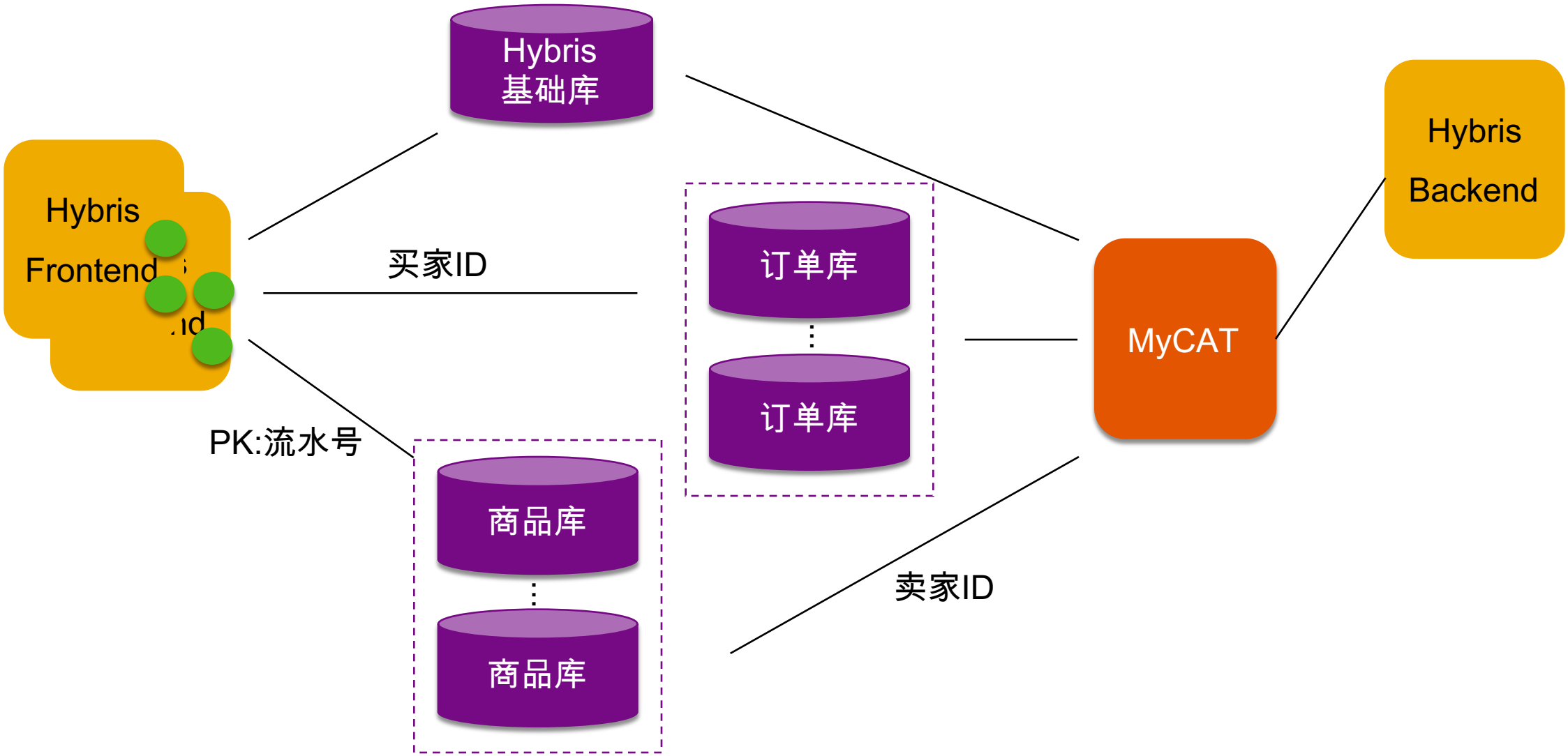
有钱买个高端数据库！

Oracle 12.2 据说支持Sharding

PoC deployment



优化部署方案


Hybris
基础库
Hybris
Frontend
买家ID
订单库
订单库
MyCAT
Hybris
Backend
PK:流水号
商品库
商品库
卖家ID

还能不能自由自在Join了

第一原则: 能不切分尽量不要切分。
第二原则: 如果要切分一定要选择合适的切分规则,提前规划好。
第三原则: 数据切分尽量通过数据冗余或表分组(Table Group)来降低跨库Join的可能。
第四原则: 由于数据库中间件对数据Join实现的优劣难以把握,而且实现高性能难度极大,业务读取尽量少使用多表Join。

算法：E-R Join, Share Join

复制：MyCAT 全局表，DRDS 小区广播表

Sharding key

五个维度考量：数据均衡度、事务边界因素、常用查询效率、异构索引、简单性策略

订单

流水号	订单ID	买家ID	卖家ID	创建时间	订单状态	销售渠道				

购物车

流水号	订单ID	买家ID	卖家ID	创建时间	订单状态	销售渠道				

商品

流水号	商品ID	商品编号	品类ID	商品编码	商家ID	销售渠道	状态	baseProd		

分表查询优化

异构复制

Table_bid
buyerID % 4

bizOrderID	buyerID	sellerID	content
0	0	1	床上用品
1	0	2	路上用品
2	0	3	销售路由器
3	0	4	中文书籍
4	0	5	电脑
8	4	0	桌面

bizOrderID	buyerID	sellerID	content
5	1	0	ipad

bizOrderID	buyerID	sellerID	content
6	2	0	笔记本

bizOrderID	buyerID	sellerID	content
7	3	0	铅笔

Table_sid
sellerID % 4

bizOrderID	buyerID	sellerID	content
5	1	0	ipad
6	2	0	笔记本
7	3	0	铅笔
8	4	0	桌面
3	0	4	中文书籍

bizOrderID	buyerID	sellerID	content
0	0	1	床上用品
4	0	5	电脑

bizOrderID	buyerID	sellerID	content
1	0	2	路上用品

bizOrderID	buyerID	sellerID	content
2	0	3	销售路由器

新的挑战

性能问题

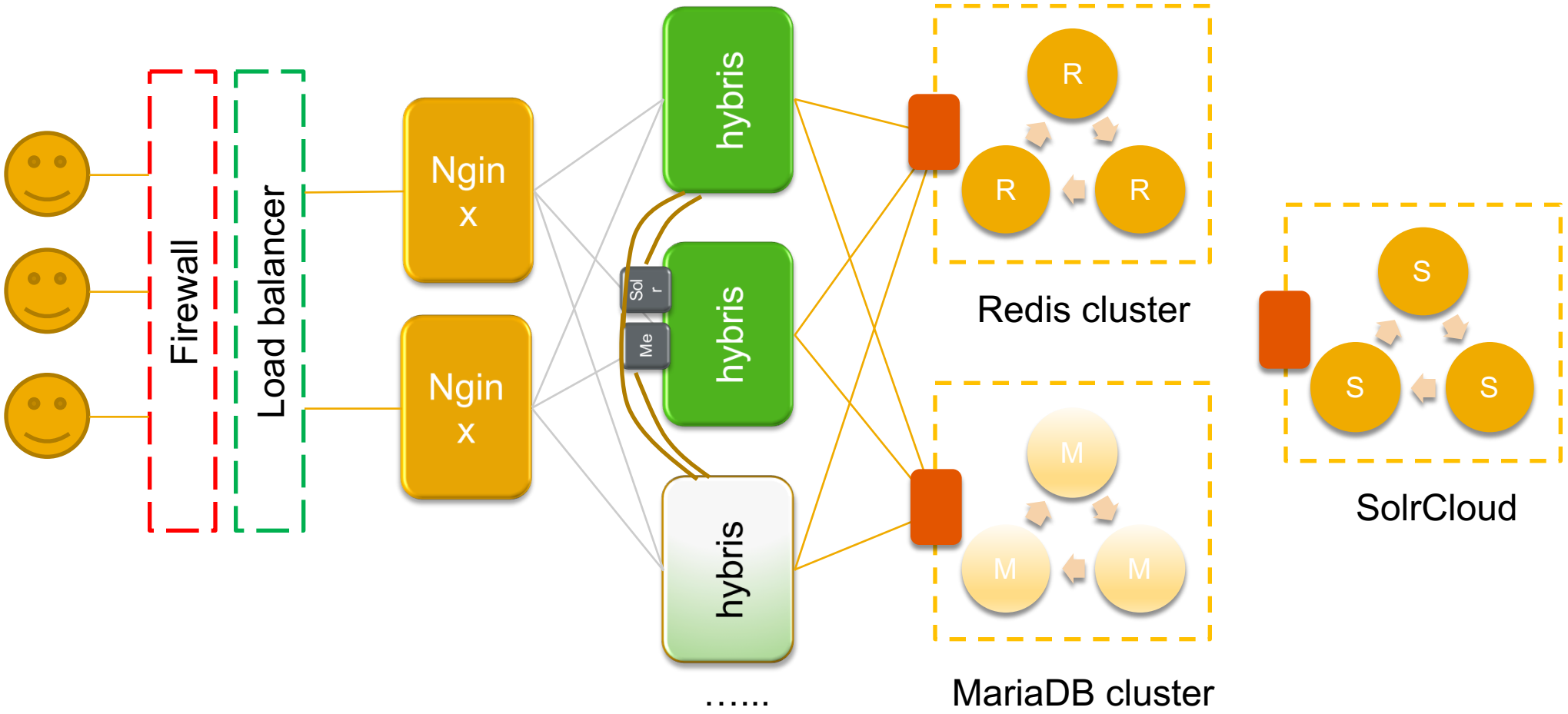
事务问题

迁移扩容问题

代码问题

业务问题

Scalable architecture diagram



技术白皮书大纲 white paper TOC

典型**hybris**电商应用架构

大规模互联网与电商应用扩展原则

应用层无状态化与服务化

数据层读写分离与分库分表

- **hybris**自带读写分离方案与测试
- **hybris**分库分表方案与测试
- **MyCat**中间件方案与测试

分布式搜索与**SolrCloud**

hybris大规模可扩展架构推荐实践

测试准备中

SAP hybris

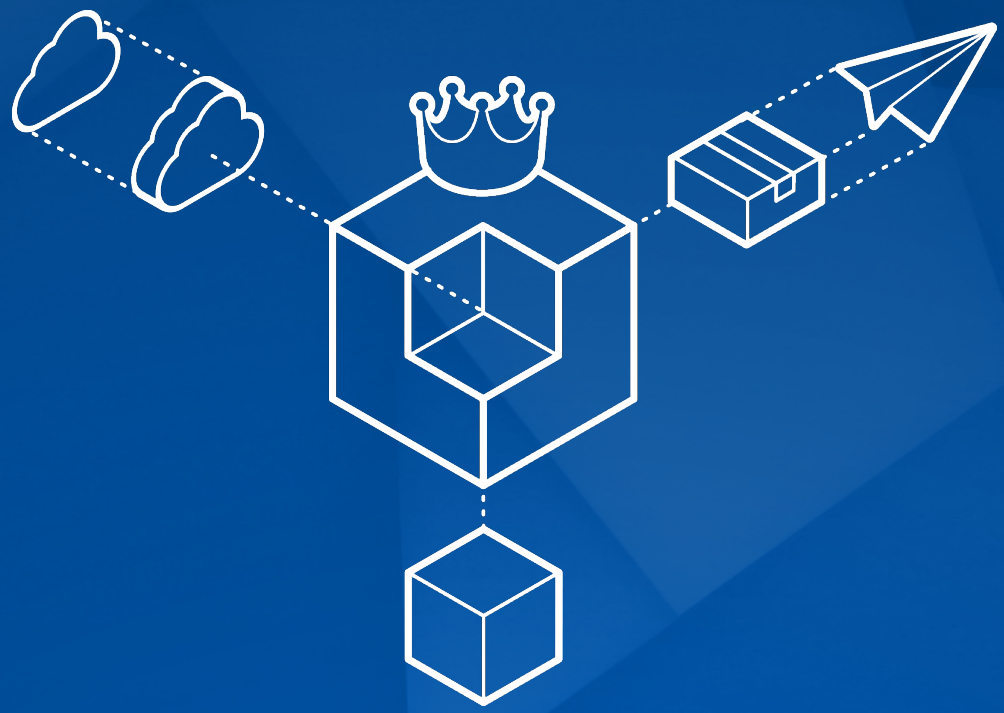
MyCat开源社区

Accenture CDC

联合发布

参考资料

1. 阿里分布式数据库实践
2. DRDS开发手册
3. DRDS最佳实践
4. MyCAT系列文档
5. MariaDB系列文档
6. Oracle技术嘉年华2015



THANK YOU