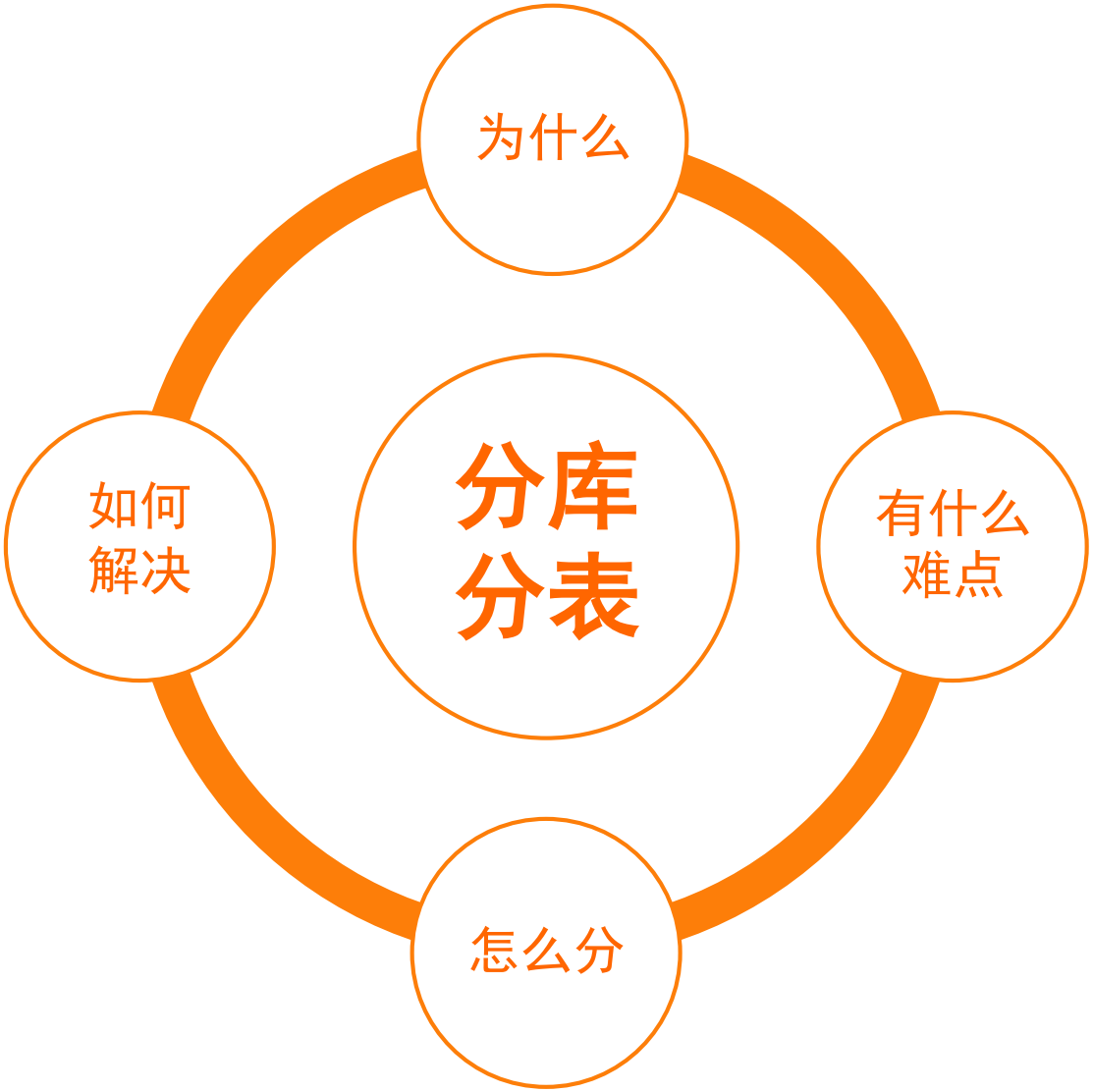
京东金融
杭州研发中心

# 基于客户端的分库分表实践

# 目录

- 分库分表理论
- CDS实践
- 如何接入CDS

# 分库分表理论



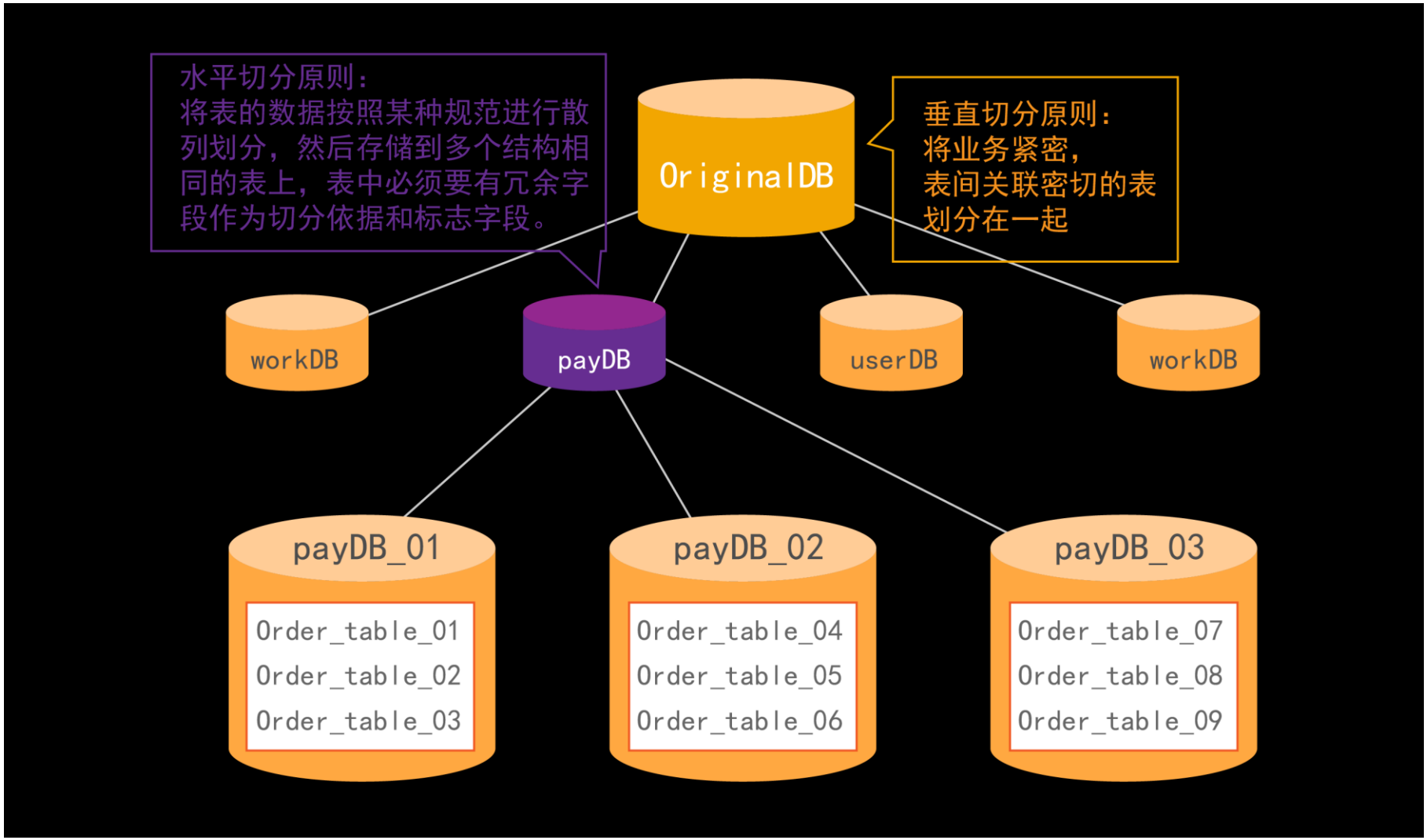
# 分库分表理论

- 为什么要分库分表?
  1. 业务数据量剧增
  2. 访问并发量剧增
  3. 业务系统可用性需求

# 分库分表理论

- 如何选择拆分策略?
  1. 垂直拆分
  2. 水平拆分
  3. 两者结合

# 分库分表理论


水平切分原则：
将表的数据按照某种规范进行散列划分，然后存储到多个结构相同的表上，表中必须要有冗余字段作为切分依据和标志字段。
OriginalDB
垂直切分原则：
将业务紧密，
表间关联密切的表划分在一起
workDB
payDB
userDB
workDB
payDB_01
Order_table_01
Order_table_02
Order_table_03
payDB_02
Order_table_04
Order_table_05
Order_table_06
payDB_03
Order_table_07
Order_table_08
Order_table_09

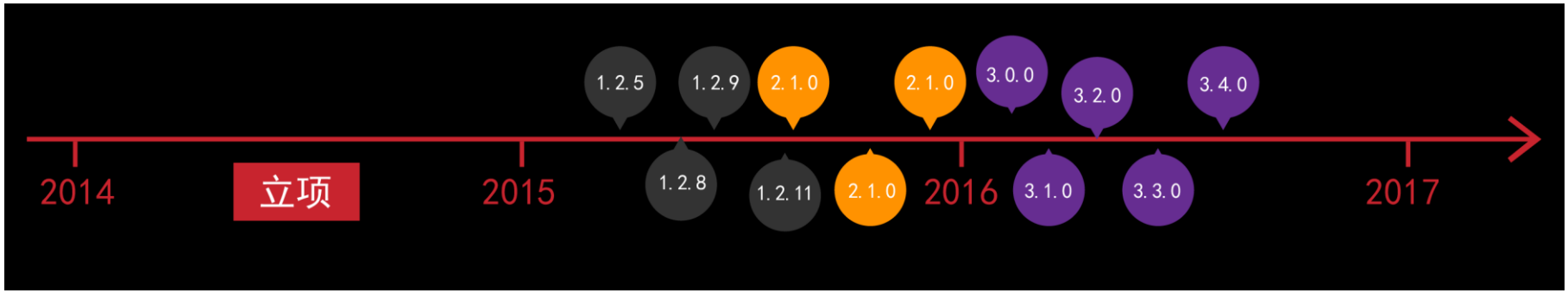
# 分库分表理论

- 分库分表带来了哪些难题？
  1. 分布式事务
  2. 复杂SQL查询
  3. 跨库跨表Join
  4. 数据运维

# 如何解决？

## Completed Database Sharding

# CDS RoadMap

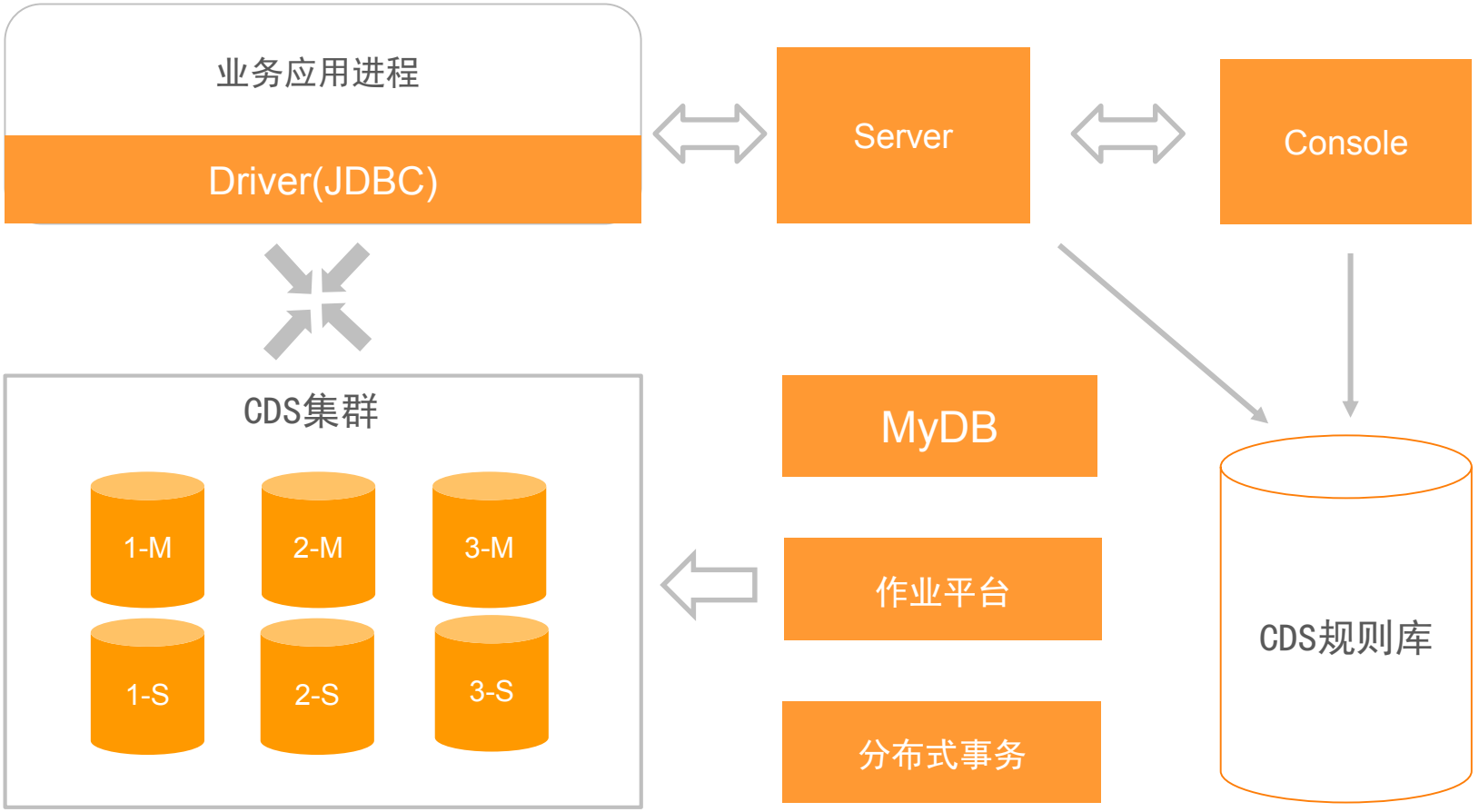



❶ 分库分表，读写分离，Failover。

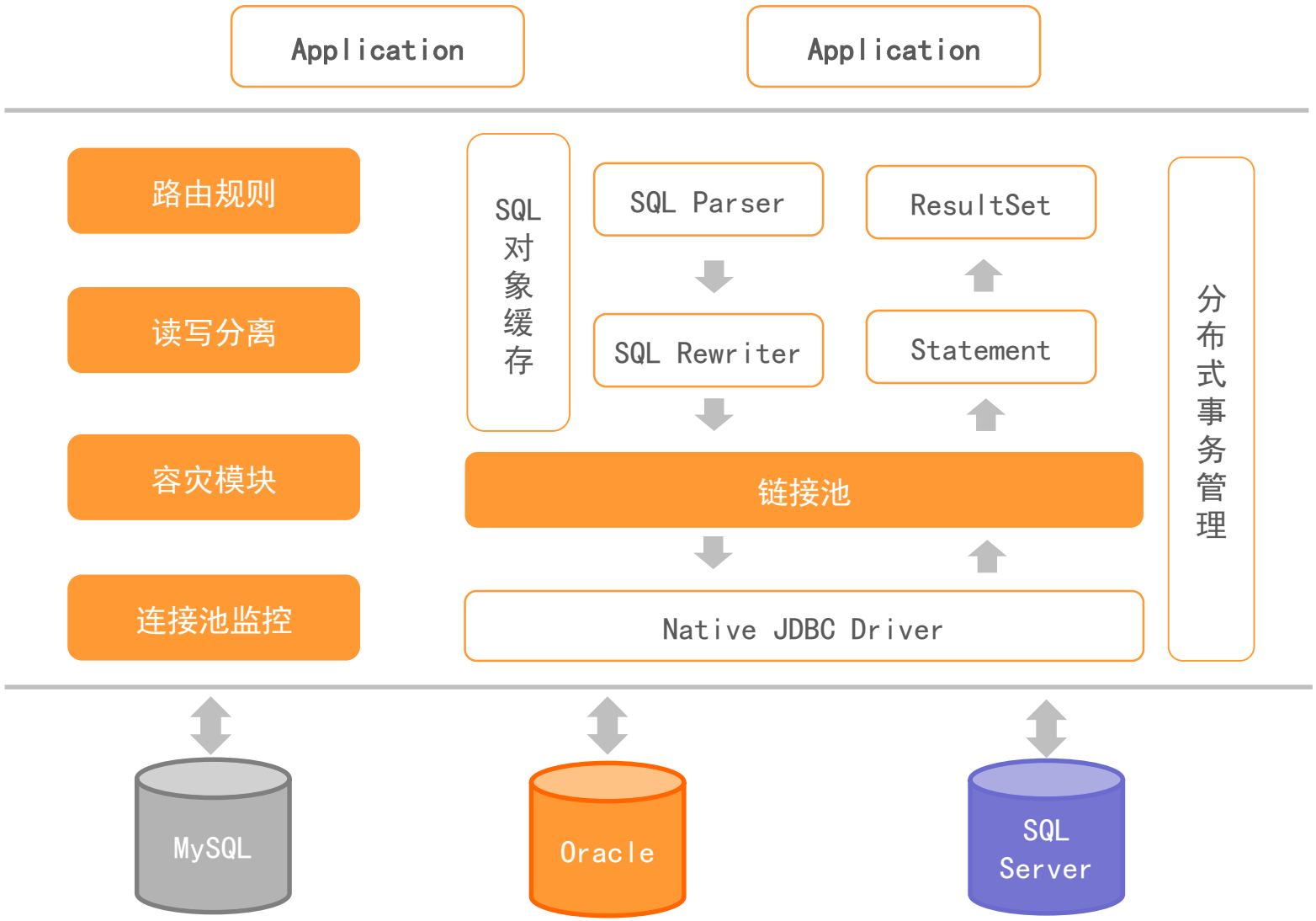
❷ 运维监控，自定义路由，级联路由，重读双写，集群配置在线推送，弱XA事务支持，作业平台，离线数据集中平台。

❸ 优化sql解析模块，丰富sql语法，groovy路由算法，数据管道服务，分布式补偿事务服务。

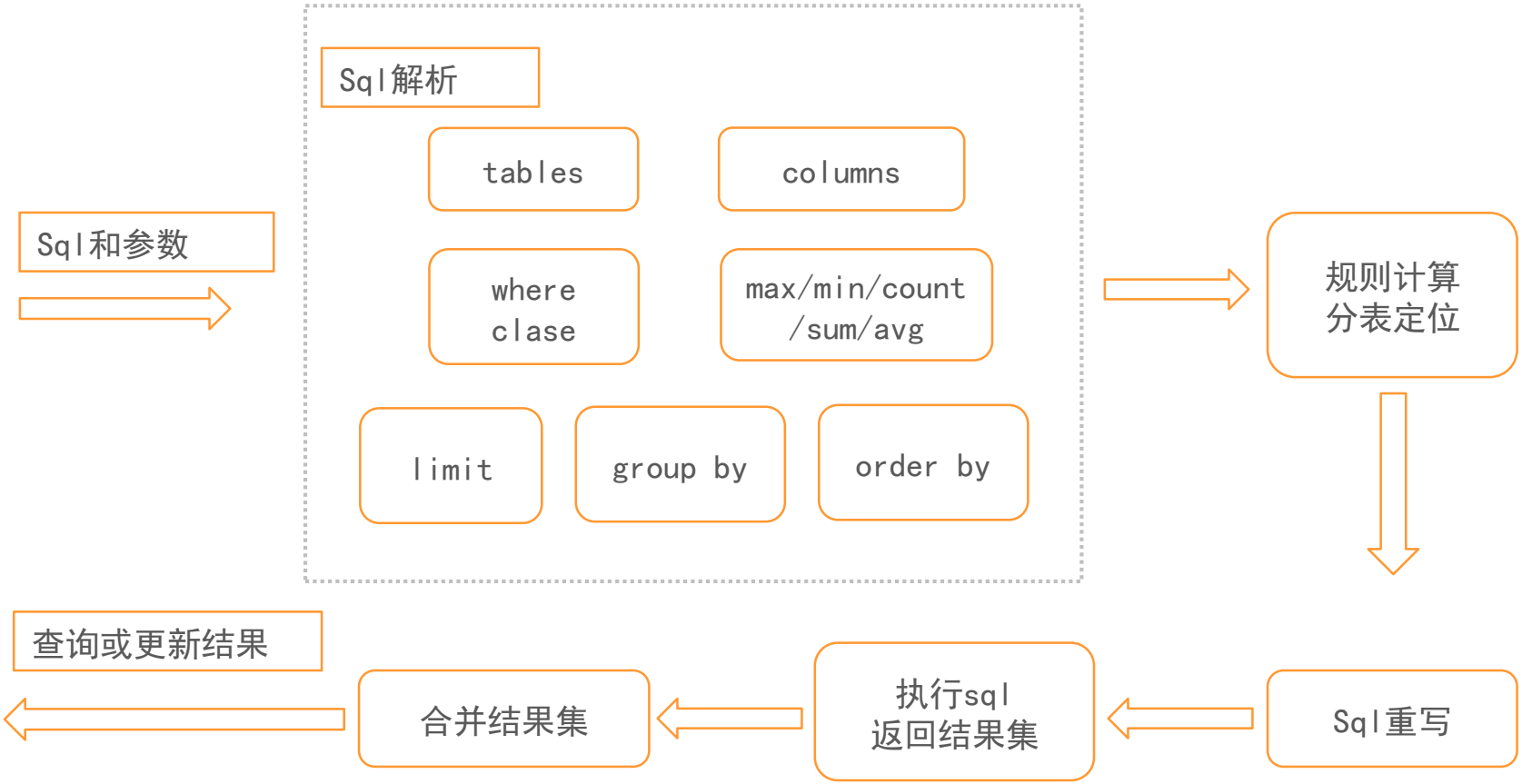
# CDS系统架构


业务应用进程
Driver(JDBC)
Server
Console
CDS集群
1-M
2-M
3-M
1-S
2-S
3-S
MyDB
作业平台
分布式事务
CDS规则库

# CDS驱动架构


Application
Application
路由规则
读写分离
容灾模块
连接池监控
SQL
对
象
缓
存
SQL Parser
SQL Rewriter
ResultSet
Statement
链接池
Native JDBC Driver
分布式事务管理
MySQL
Oracle
SQL
Server

# SQL查询流程


Sql解析
tables
columns
where clase
max/min/count /sum/avg
limit
group by
order by
Sql和参数
规则计算 分表定位
Sql重写
执行sql 返回结果集
合并结果集
查询或更新结果

# CDS集群元素

- 群组类型
  1. 全局组
  2. 工作组
  3. 空备组
  4. 外挂组
  5. 重读组
- 表类型
  1. 切分表
  2. 全局表
  3. 孤立表
- 切分键类型
  1. 级联切分键
  2. 复合切分键
  3. 查询切分键

# CDS集群结构

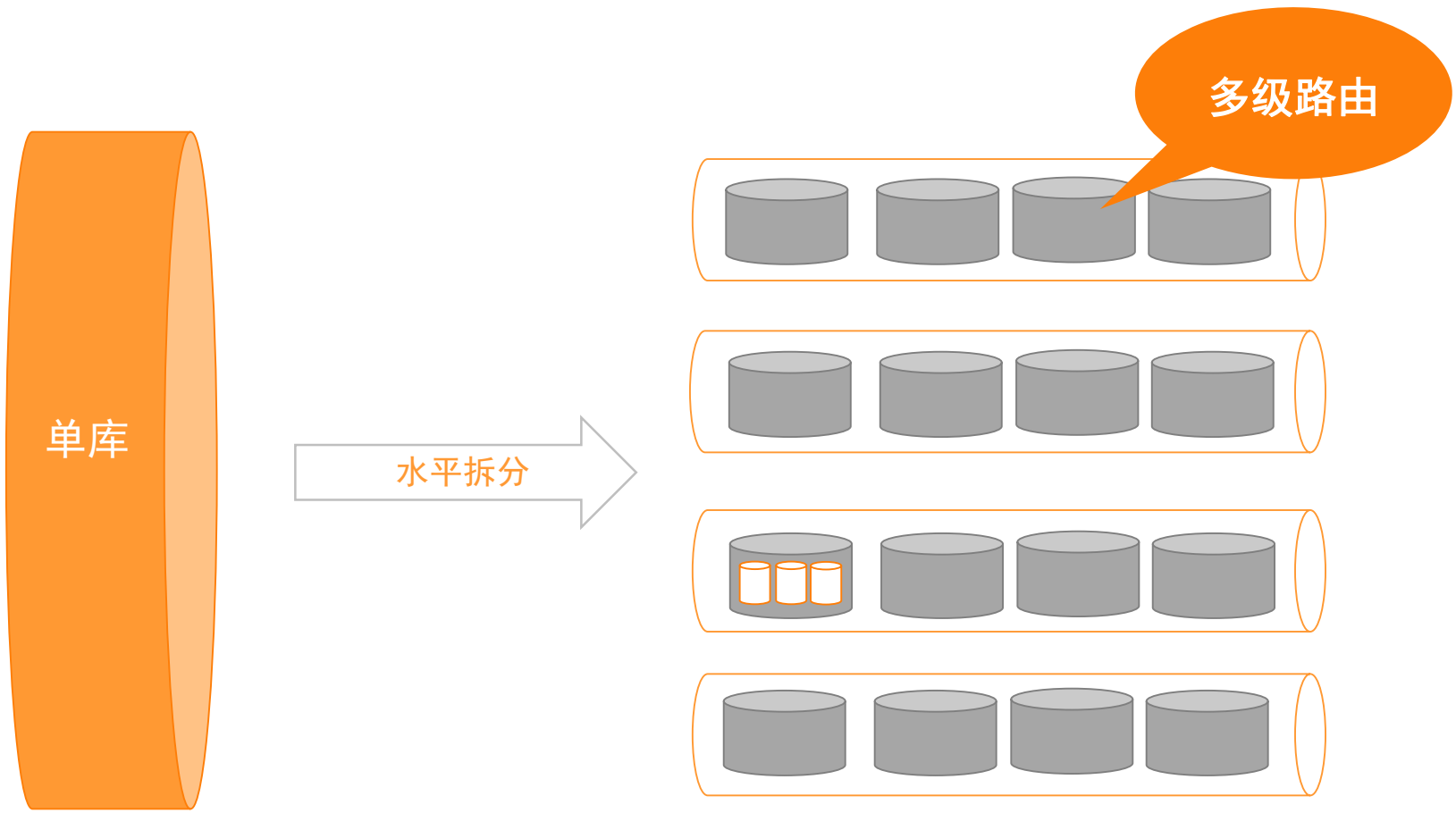

## CDS集群

**全局表：**每个工作组都存在相同的数据备份(比如. 城市、类别）

**切分表：**插入数据时根据切分键插入到不同分表（比如. 业务流水表）

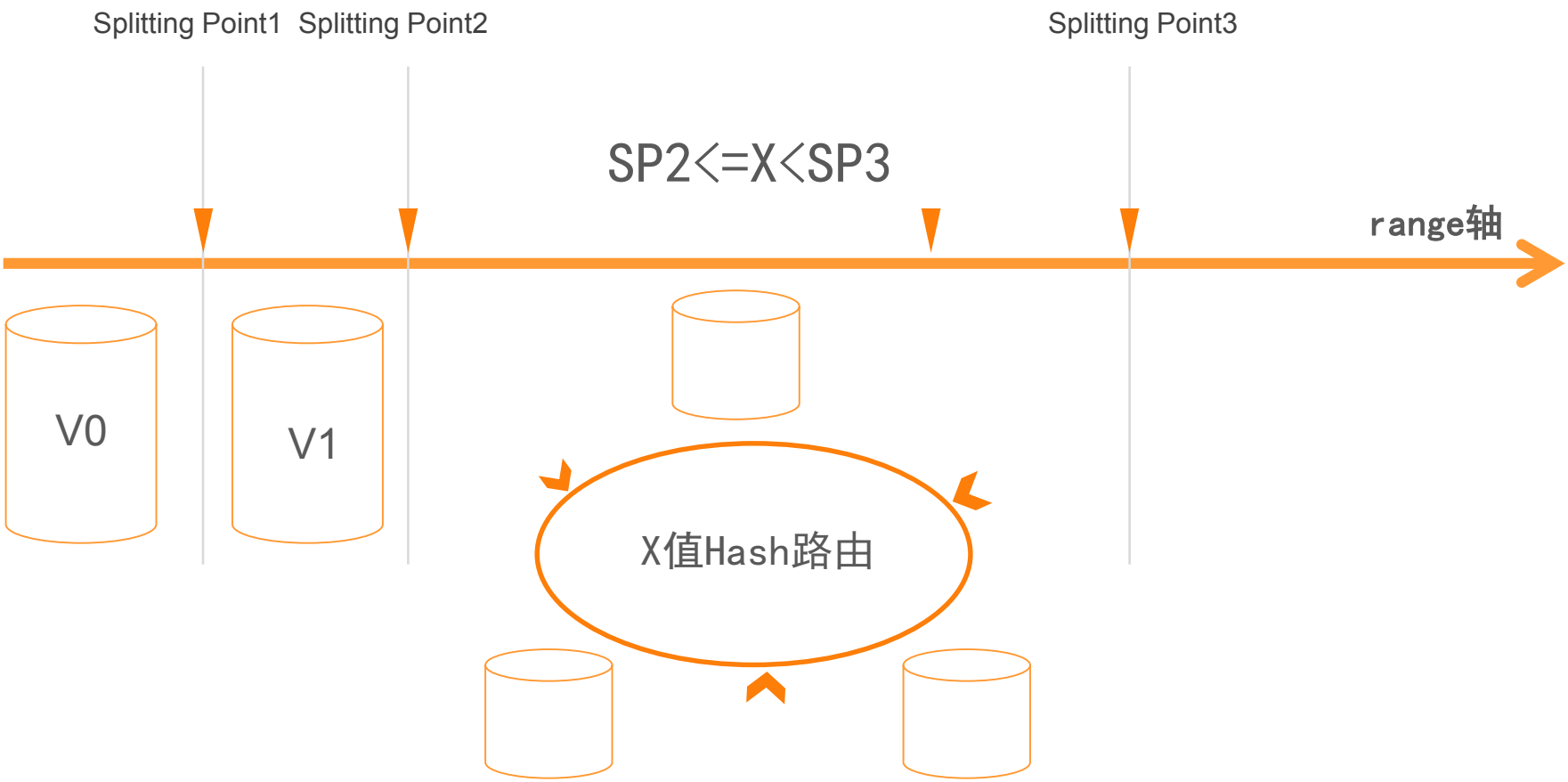
**孤立表：**可以存放在任意工作组包含数据表（比如. 配置表）

# CDS水平拆分



# 级联路由策略

- Range路由策略 + Hash路由策略



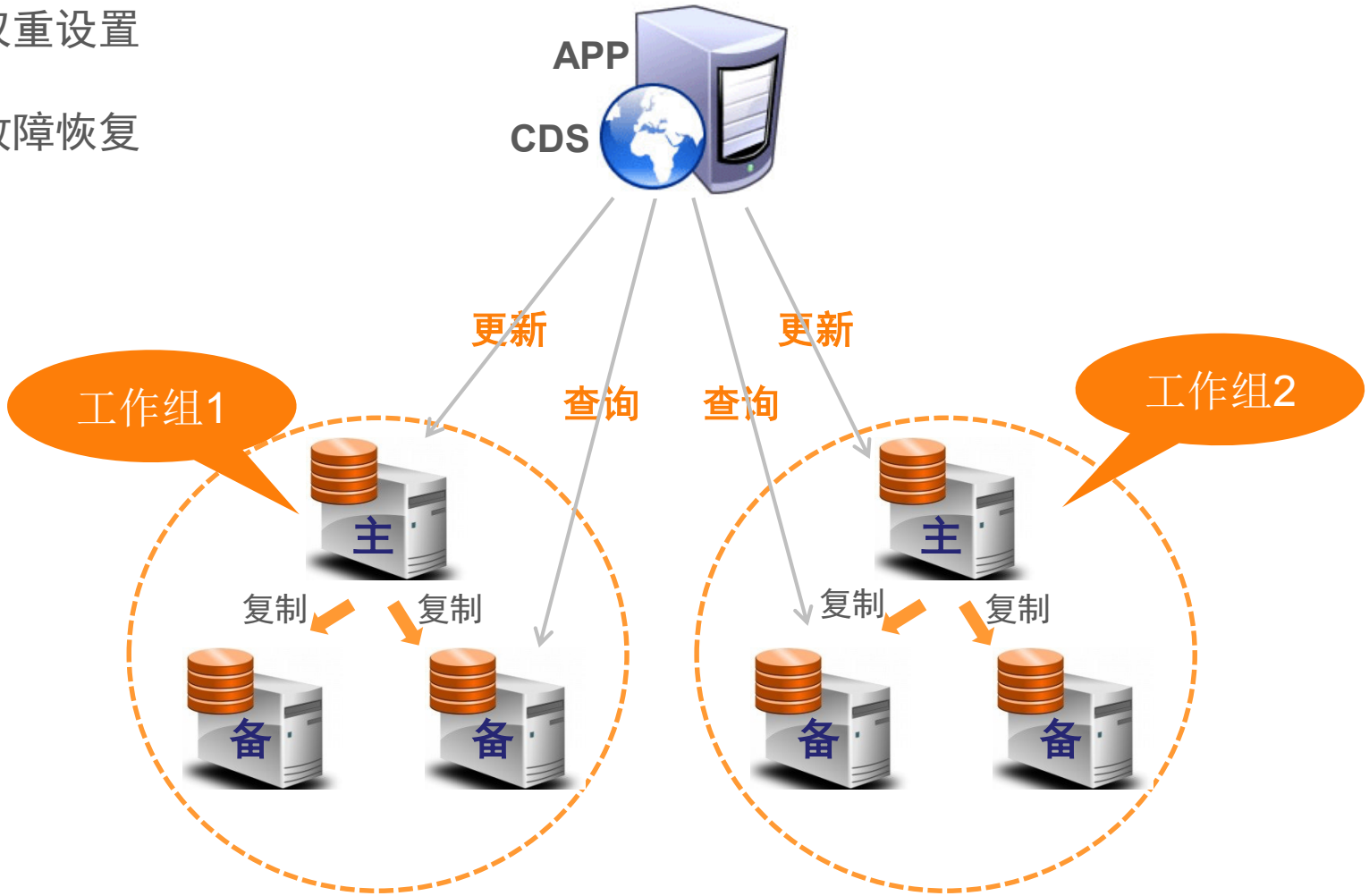
# CDS基本特性

- 分库分表
  1. 分区规则支持：Hash, Range, List，自定义（groovy）
  2. 支持oracle、mysql、sql server数据库
  3. 支持复合切分键
  4. 支持级联切分键
  5. 支持DDL变更
  6. 支持Sum/Max/Min/Count/Avg聚合函数
  7. 不支持：复杂子查询、更新切分键字段

- 非分库分表
  1. 支持存储过程
  2. 支持大部分DML、DDL

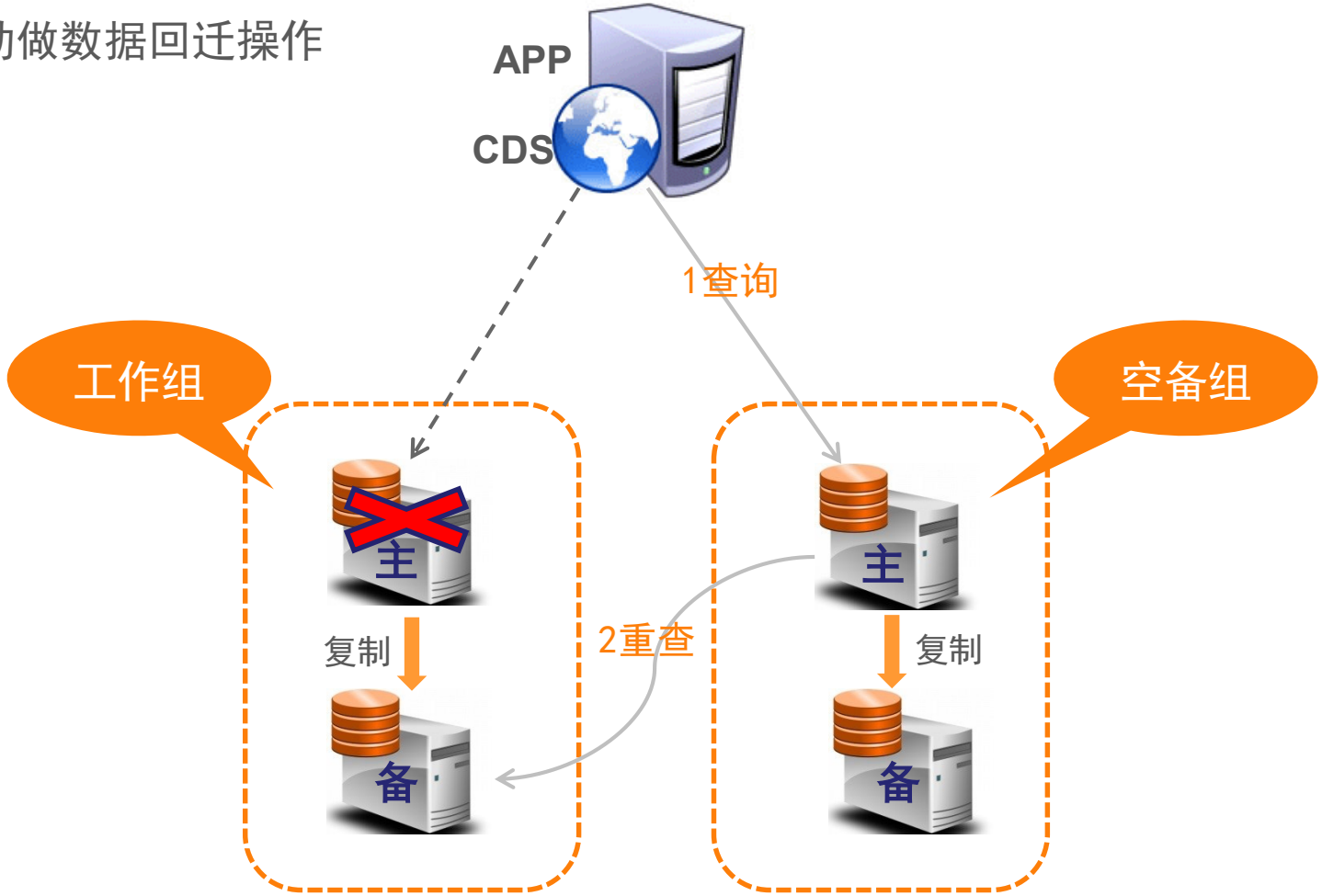
# CDS特性--读写分离

1. 负载均衡
2. 权重设置
3. 故障恢复



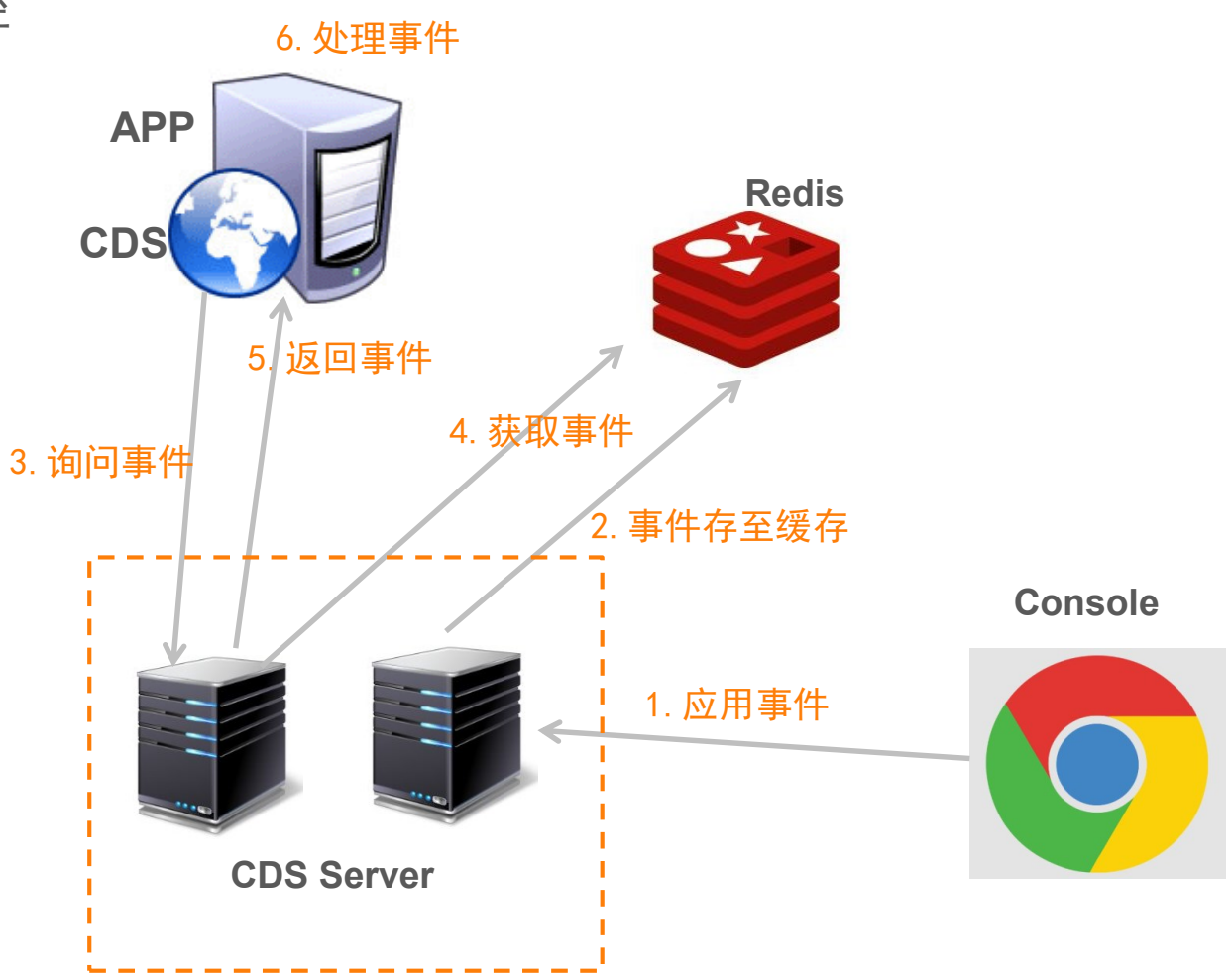
# CDS特性—Failover

1. 适用于流水型业务表，记录变更与查询不依赖之前的信息
2. 主库故障由生产DBA手动触发推送事件，将新请求指向failover库，主库恢复后需要手动做数据回迁操作



# CDS特性--事件推送

1. 连接池监控
2. 应用进程信息监控
3. 集群配置变更



# CDS特性--Sequence

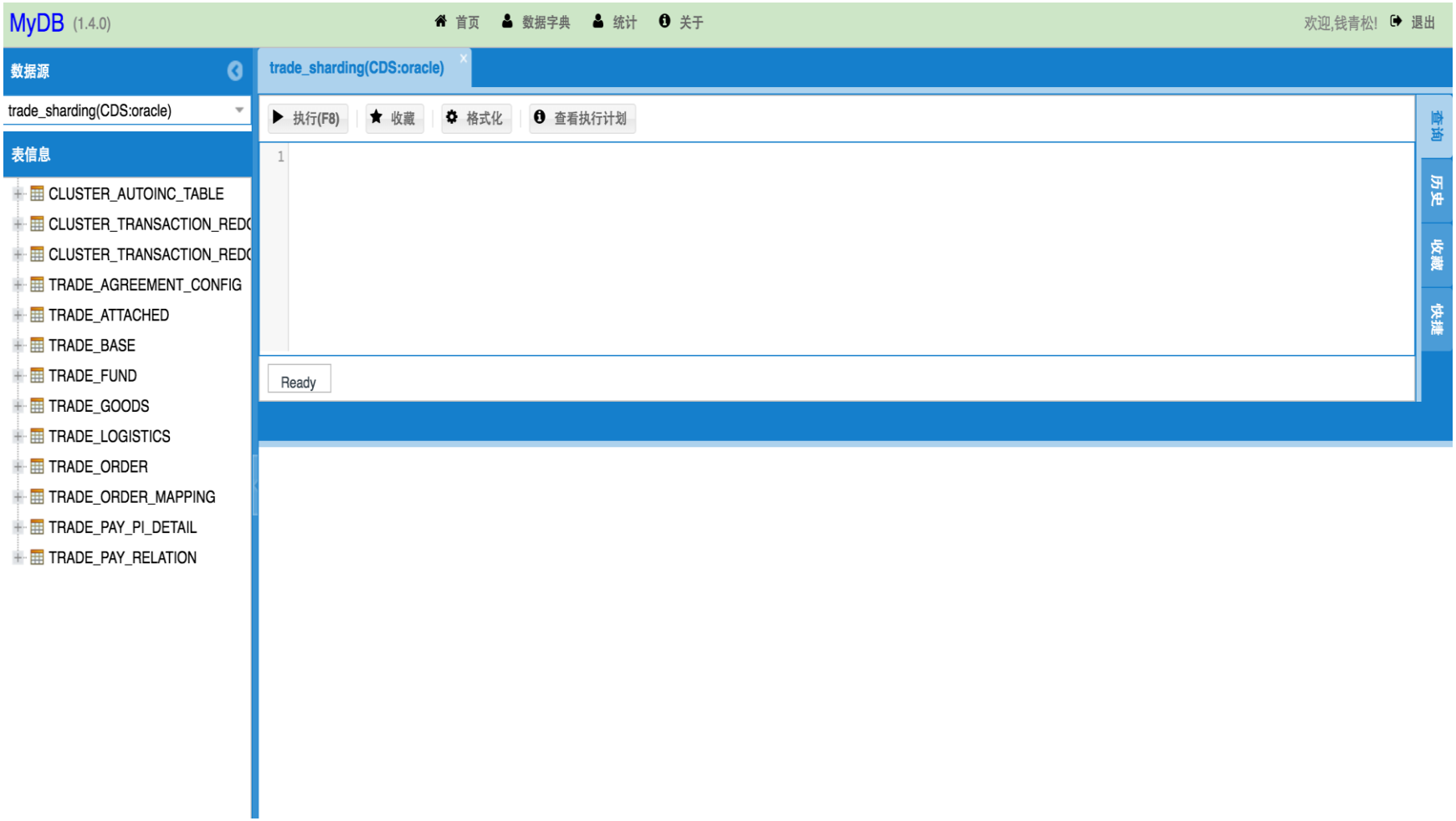
- 全局Sequence
  1. 全局唯一
  2. 不保证全局有序
  3. 单点问题
- 分表Sequence
  1. 解决单点问题
  2. 与分表一一对应
  3. 唯一性由业务保证

# CDS工具集

- MyDB
- 环境检查工具
- 作业平台
- 数据管道服务
- 分布式补偿事务
- 数据集中平台
- …… ……

# MyDB

CDS数据库源在线查询，订正，统计，导出，支持工单和复杂用户权限控制



# 作业平台

- 海量数据迁移
- 增量数据同步
- 自动建表
- Failover数据回迁
- 批量数据导出
- 集群数据库对象比对
- 索引表初始化
- ……

# 数据管道


业务数据
业务库
BinLogs
数据管道服务
Binlog读取解析
过滤、规整、分发
数据同步服务
分布式事务补偿
事务消息表WORKER
其它应用场景

# 分布式事务

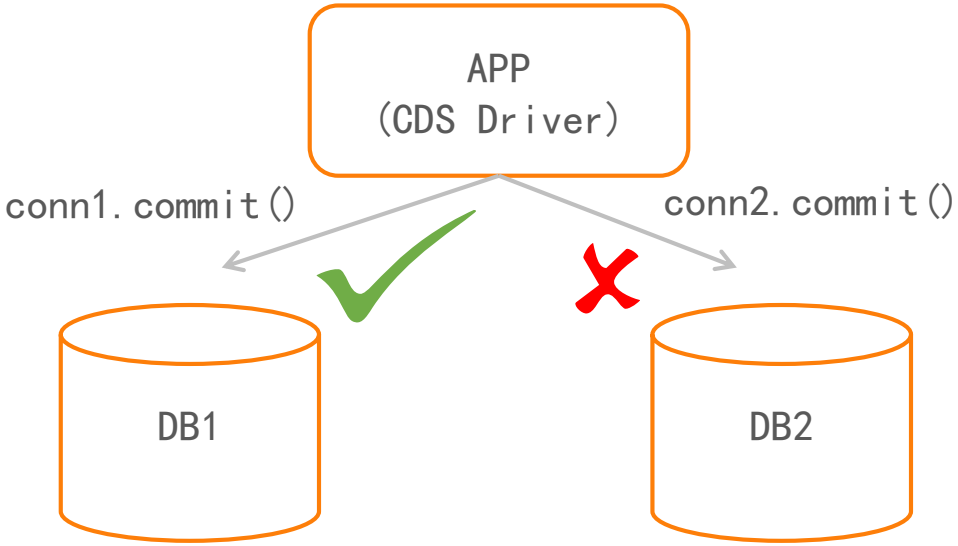
- 分布式系统事务处理的三种方式：
  1. 分布式事务
  2. 基于Best Efforts 1PC模式的事务
  3. 事务补偿机制

- 小结：
  1. 分布式事务，最严格的事务实现，性能是个大问题；
  2. Best Efforts 1PC模式，性能与事务可靠性的平衡，支持系统水平伸缩，大多数情况下是最合适的选择；
  3. 事务补偿机制，最终一致，牺牲实时一致性，获得最大的性能回报。

# CDS事务补偿

- 基于BASE模型，通过解析Mysql的binlog来实现事务补偿，对用户完全透明，不一致时间窗口控制在秒级；
- 最终状态为事务失败状态，业务能感知事务失败异常；

# CDS事务补偿

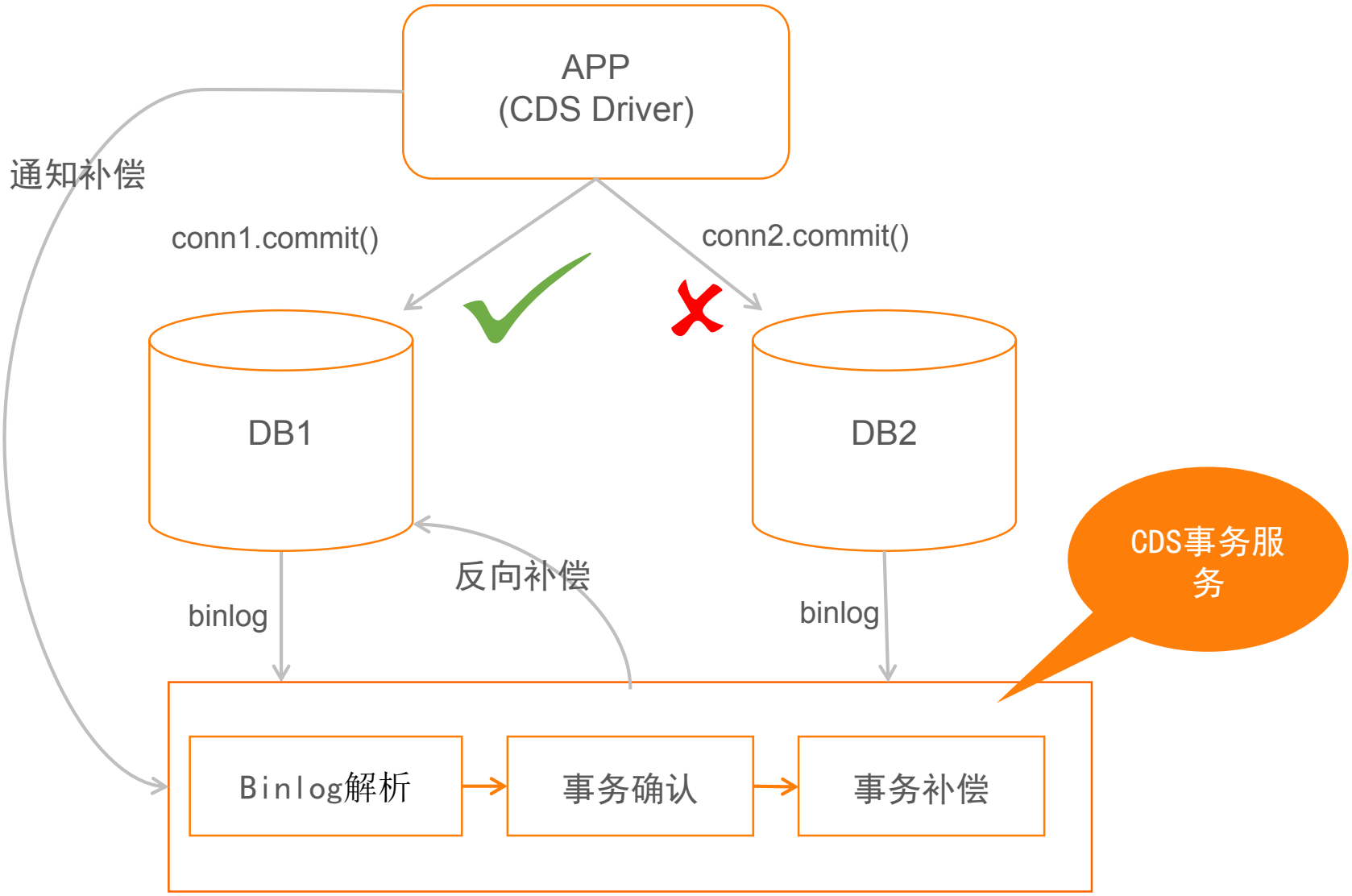



- 问题：

  ❶ 补偿哪个binlog事务?

  ❷ 怎么建立conn1.commit()和conn2.commit()的联系

# CDS事务补偿


APP
(CDS Driver)
通知补偿
conn1.commit()
conn2.commit()
DB1
DB2
反向补偿
binlog
binlog
CDS事务服务
Binlog解析
事务确认
事务补偿

# CDS事务补偿

```
try{
        connection1.insert、update、delete...
        connection2.insert、update、delete...
        ... ...
        try{
                insert  cluster_xa_1
                insert  cluster_xa_2
                connection1.commit();
                connection2.commit();
         }
         catch(Exception e){
                notifyTxServer(); //通知事务服务
                .....
         }
}catch(Exception e){
        connection1.rollback();
        connection2.rollback();
}
```

CDS Driver向每个分库cluster_xa表插入一条相同记录，业务服务完全透明

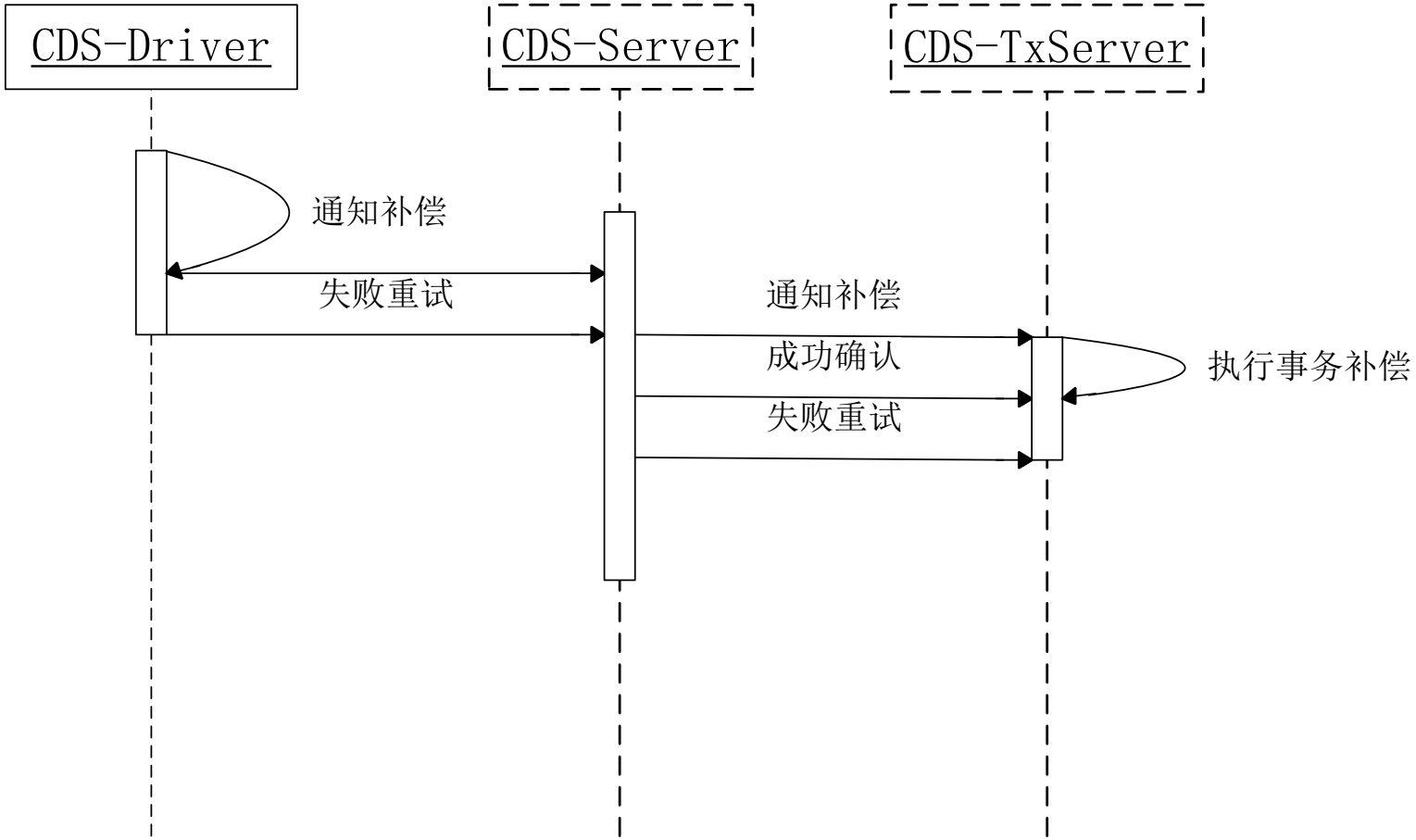
# CDS事务补偿

- Cluster_XA表

  1. xa_uuid：每个外围事务分配一个uuid，唯一标识分布式事务
  2. xa_dbunits：外围事务涉及到的数据库单元
  3. cur_dbunit：当前分库标识；
  4. timestamp：时间戳，用于计算超时时间

| xa_uuid | xa_dbunits | cur_dbunit | timestamp |
| --- | --- | --- | --- |
| xxx | db1,db2 | db1 | xxx |
| … … | … … | … … | … … |

# CDS事务补偿--时序图


CDS-Driver
CDS-Server
CDS-TxServer
通知补偿
失败重试
通知补偿
成功确认
失败重试
执行事务补偿

# CDS事务补偿

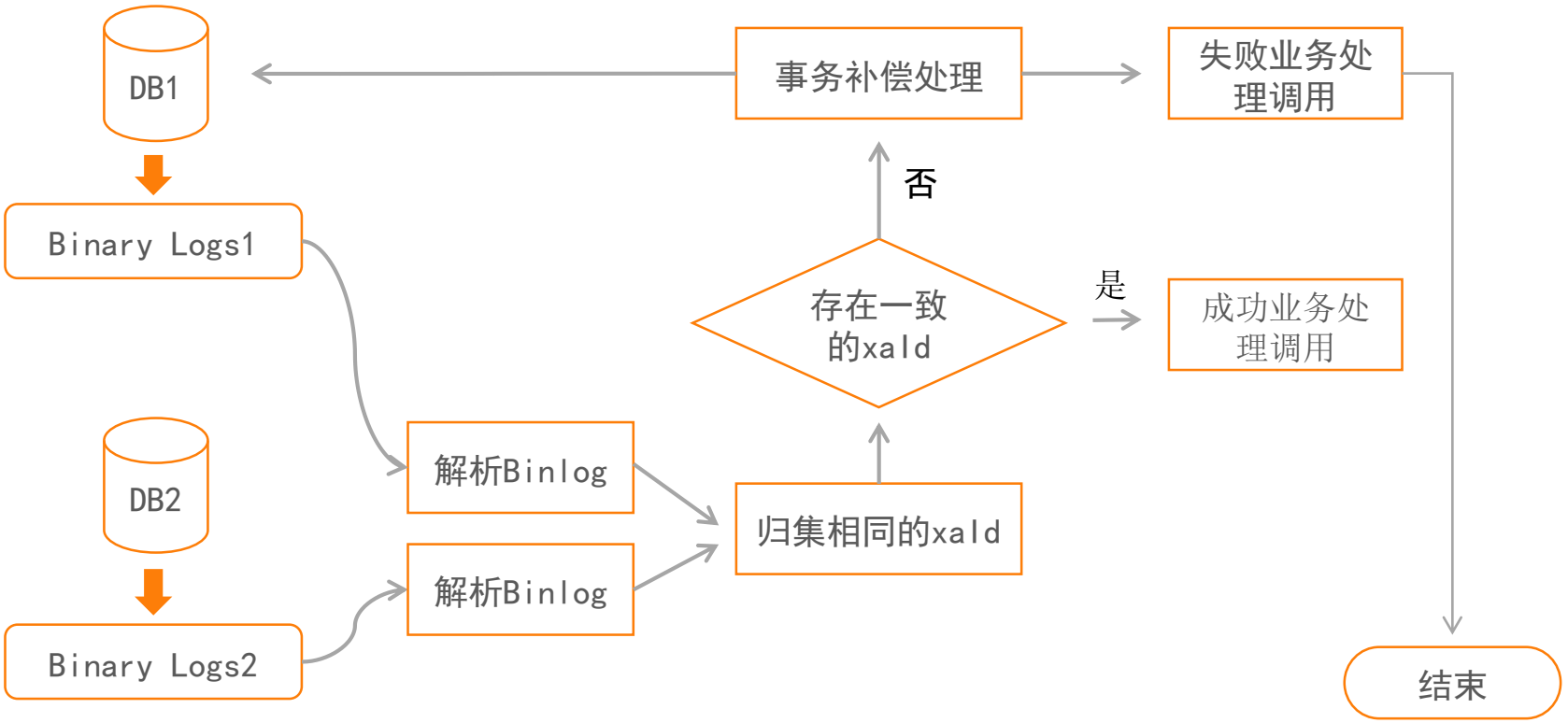
- 解决了DB2宕机事务补偿；
- 解决了到DB2网络中断，无法提交的事务补偿；

？

**进程自己宕机了，**

**而无法向TxServer发出事务补偿通知…**

# CDS事务补偿--进程宕机




说明：

- DB1提交成功了事务，会存在一条事务id,并通过cluster_xa表获悉同事务还包含DB2;
- DB2不存在相同的事务id，Binlog解析的当前时间大于DB1的xaid + 一次IO最大等待时间时，则判断DB2没有提交成功！

# CDS数据仓库

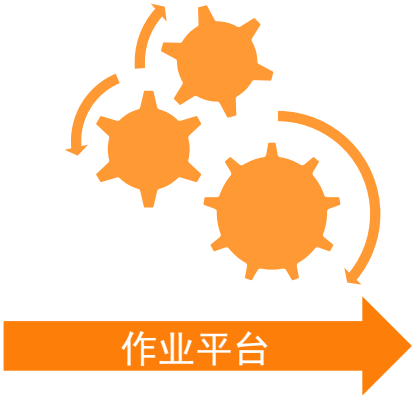
URL/JDBC

OLTP

a）海量数据存取，提高TPS

b）读写分离

c）Failover高可用



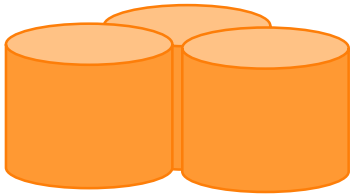

a）实时的流式数据处理

b）作业调度

c）执行监控

SOA

OFF LINE



a）离线、非实时

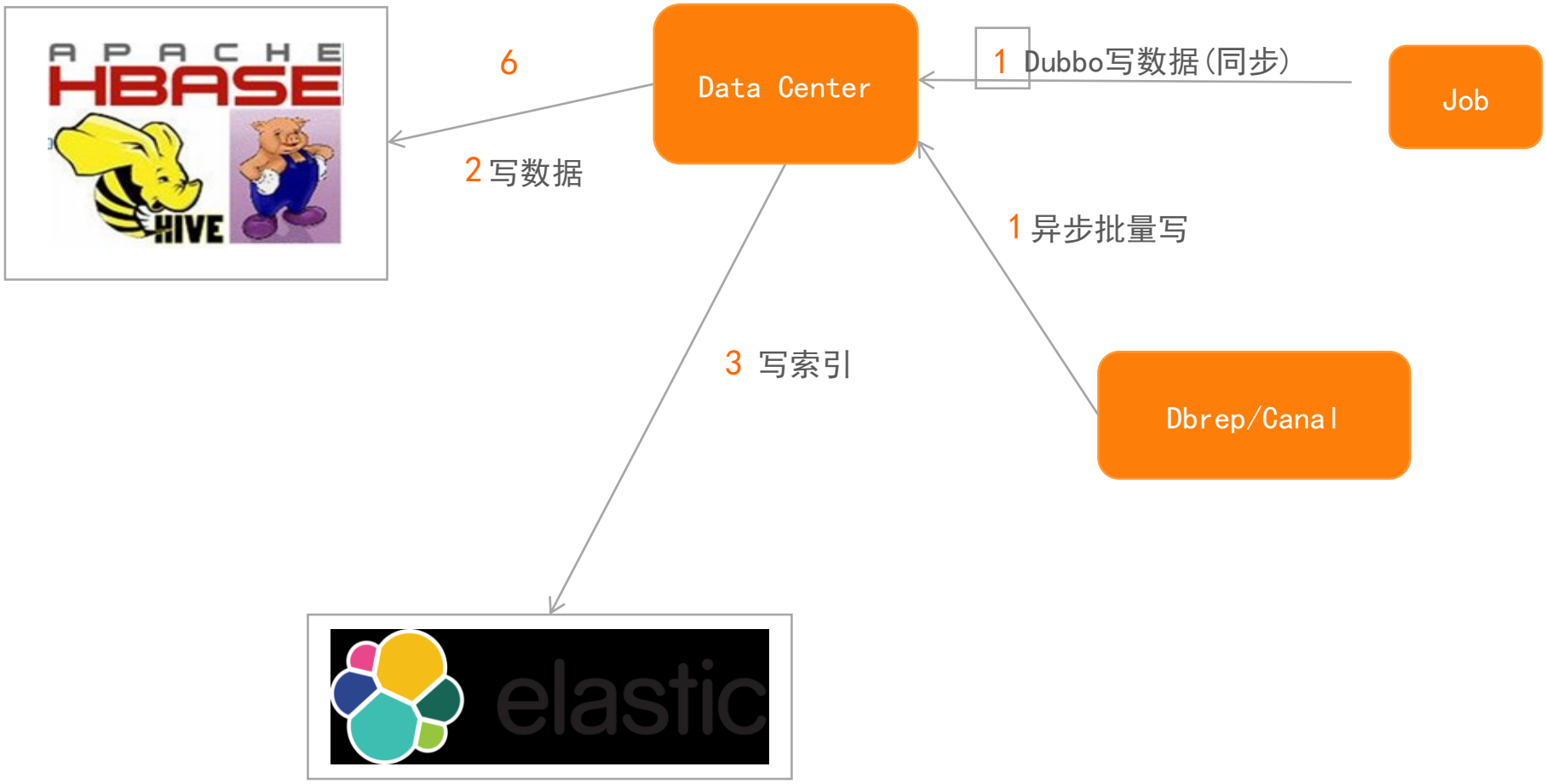
b）多维度查询

c）简单的统计

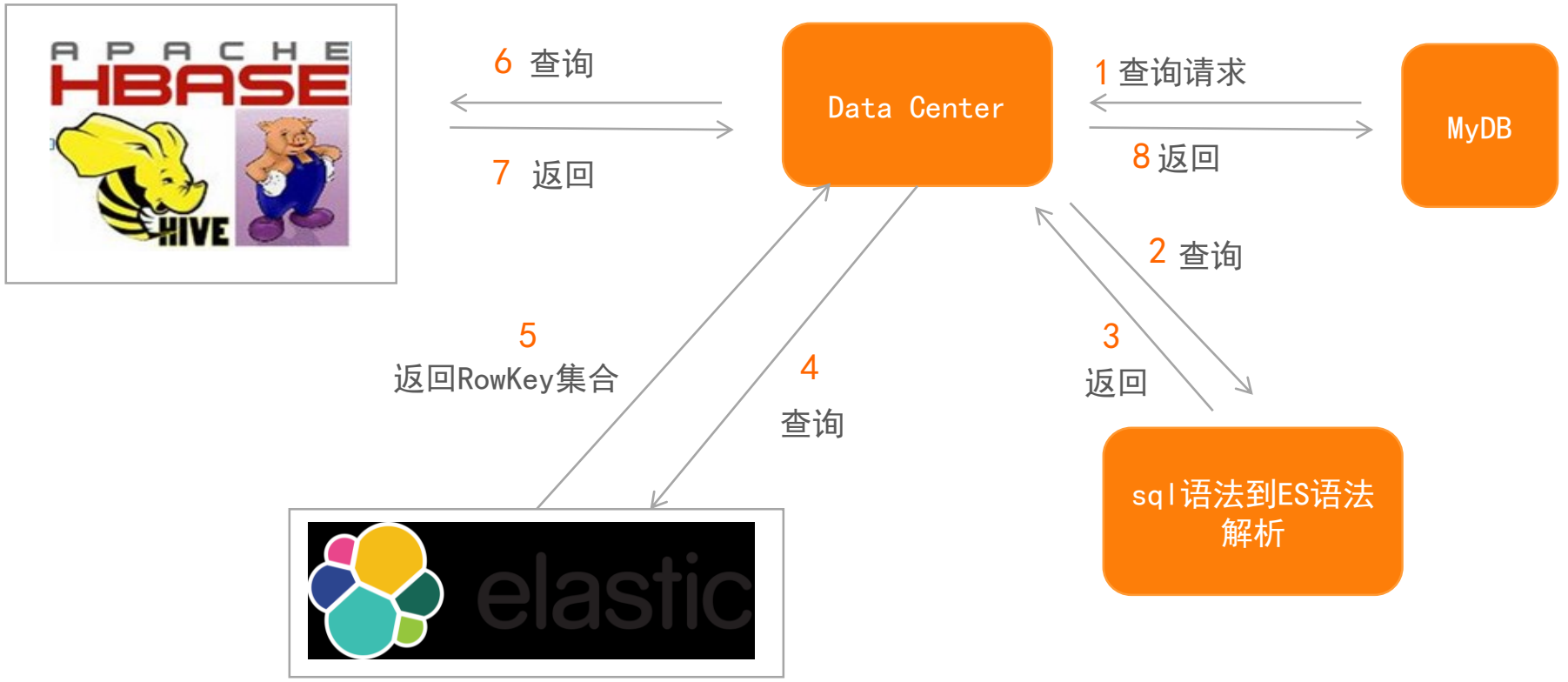
# 数据集中平台

- 离线数据查询、统计，服务于运维；
- 只关注有限时间内的数据，比如1年内；
- 索引创建，服务多维度查询
- 简单数据统计

# 数据写入


APACHE HBASE
HIVE
6
2 写数据
Data Center
1 Dubbo写数据(同步)
Job
1 异步批量写
3 写索引
Dbrep/Canal
elastic

# 数据读取(非RowKey)



# 数据读取(含RowKey)


4 查询

5 返回

Data Center

1 查询请求

6 返回

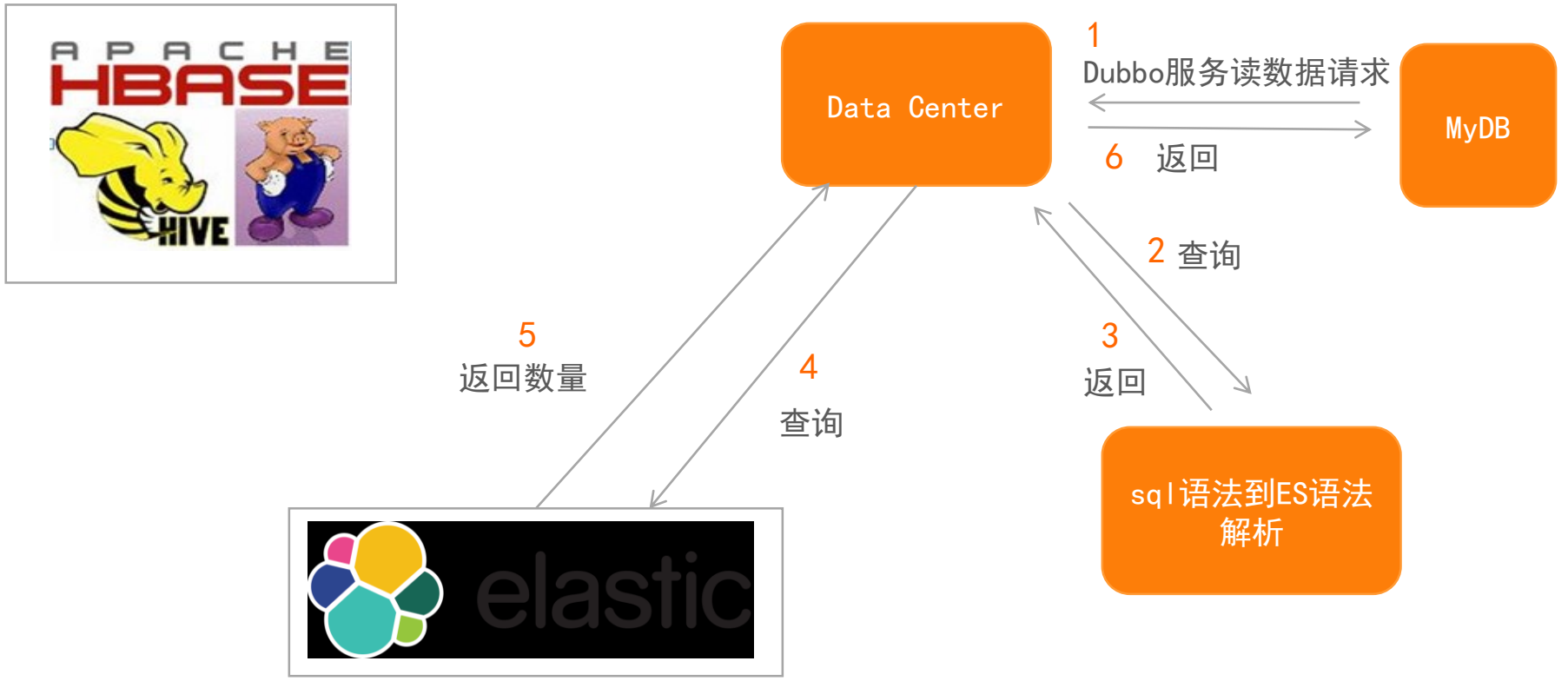
MyDB

2 查询

3 返回

sql语法到ES语法解析

# 数据读取(统计)


APACHE HBASE
HIVE
Data Center
1
Dubbo服务读数据请求
6 返回
MyDB
2 查询
3
返回
4
查询
5
返回数量
sql语法到ES语法
解析
elastic

# 欢迎使用：CDS申请接入流程

**联系方式：**

邮箱：wy-cds@jd.com

咚咚：wyxuliquan@jd.com

wywuyue@jd.com

wangzhengzheng@jd.com

# THANK YOU !