dble-开源MySQL分布式中间件实现剖析

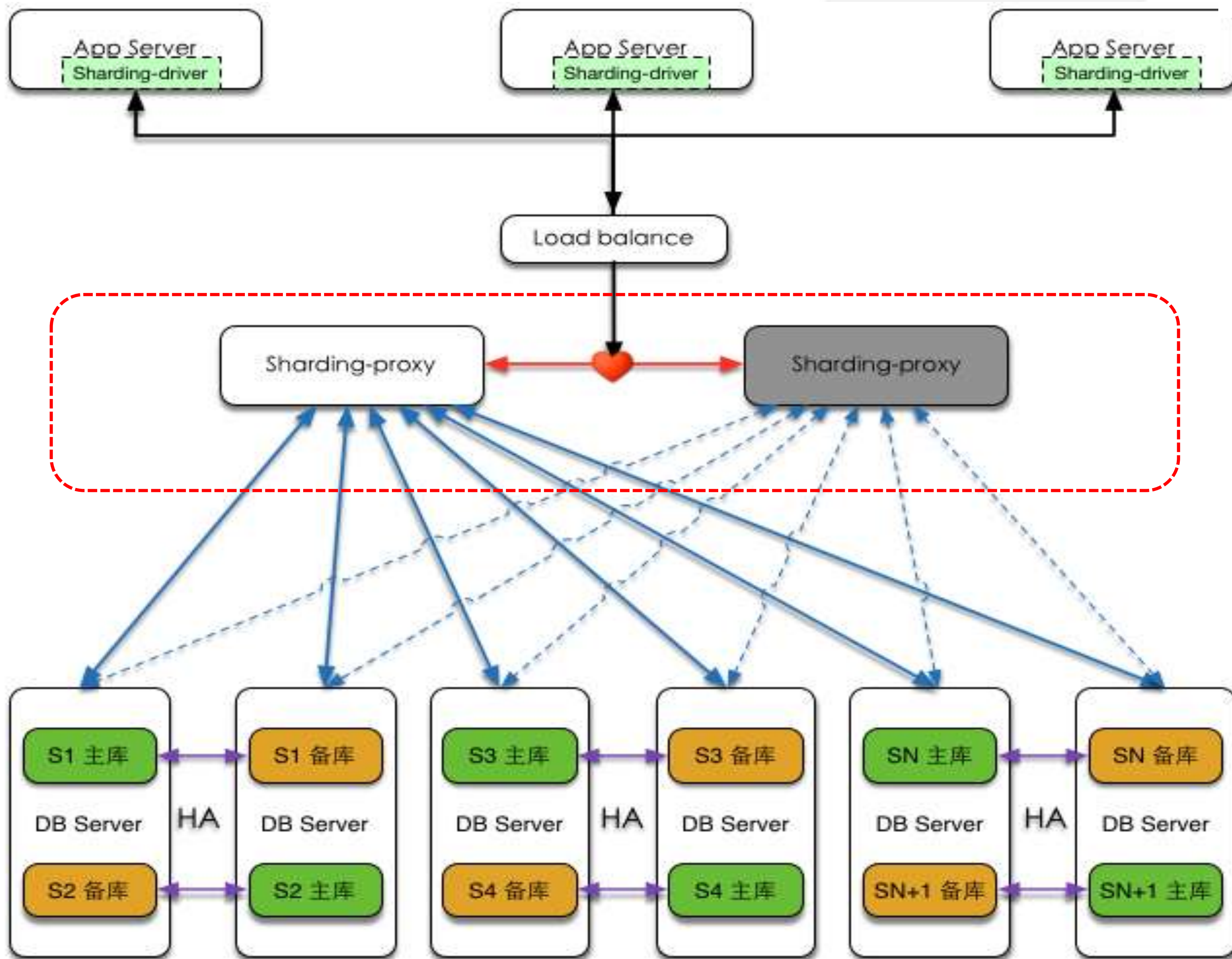
爱可生—阎虎青

ACTION 爱可生 | IT大咖说 知识分享平台

一、dble 简介
二、分布式事务实现
三、复杂查询实现
四、查询优化举例

ACTION 爱可生

分布式系统架构示意图



ACTION 爱可生 TECHNOLOGY | IT大咖说 知识分享平台

ACTION 爱可生 TECHNOLOGY

开源分布式中间件d[ou]ble

项目地址:https://github.com/actiontech/dble

ACTION TECHNOLOGY 爱可生 | IT大咖说 知识分享平台

ACTION TECHNOLOGY 爱可生

开源分布式中间件d[ou]ble

ACTION 爱可生 | IT大咖说 知识分享平台

dble是基于MyCat的企业级加强版的sharding中间件

- 优化代码结构
- Bug修复(100+)
- 核心功能完善（例：分布式事务，SQL支持度等）
- 复杂查询的支持与改进
- 非核心功能选择性裁减（例如异构数据库支持等）

项目地址:https://github.com/actiontech/dble

ACTION 爱可生

dble 功能特性

ACTION 爱可生 | IT大咖说 知识分享平台

分布式事务
- 2PC协议
- 隐式分布式识别
- 数据一致性保障
- 自动故障恢复

复杂查询
- 分布式查询计划
- 聚合函数(均值 方差)
- 聚合/矢量函数嵌套
- 跨结点JOIN
- 跨结点UNION
- 子查询(Road Map)
- 视图(Road Map)
- 查询优化

易用性
- 全局序列
- meta数据管理
- 一致性备份点
- 多维度状态信息
- 管理用户隔离
- 安全审计黑白名单
- DML权限管理
- SQL统计

兼容性
- MySQL通信协议
- SQL92标准
- 上下文同步
- 系统变量设置
- PREPARE(Road Map)

其他功能
- 读写分离
- 集群部署
- 资源池化
  连接,内存,线程
- 心跳检测

多种类型表支持
- 全局表
- 父子表
- ER表(数据分布相同)
- 非拆分表

Hint
- 指定路由
  存储过程
  insert ...select ...
  指定执行结点
  指定schema
- 指定读写节点

项目地址:https://github.com/actiontech/dble

ACTION 爱可生

分布式事务实现

ACTION 爱可生 TECHNOLOGY
IT大咖说 知识分享平台

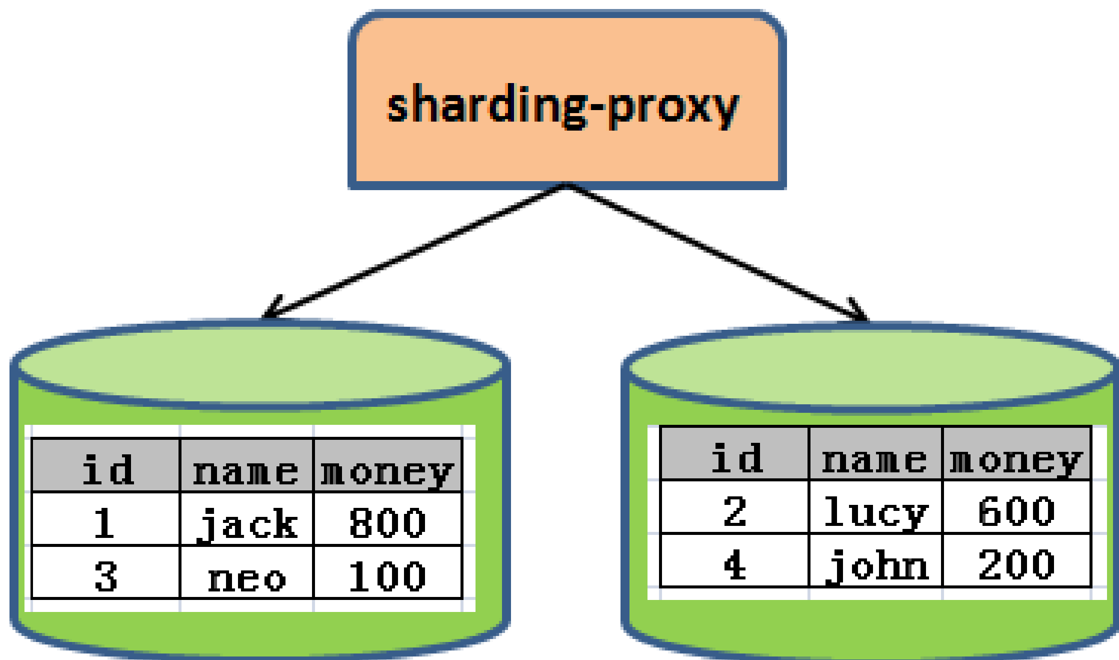
ACTION 爱可生 TECHNOLOGY

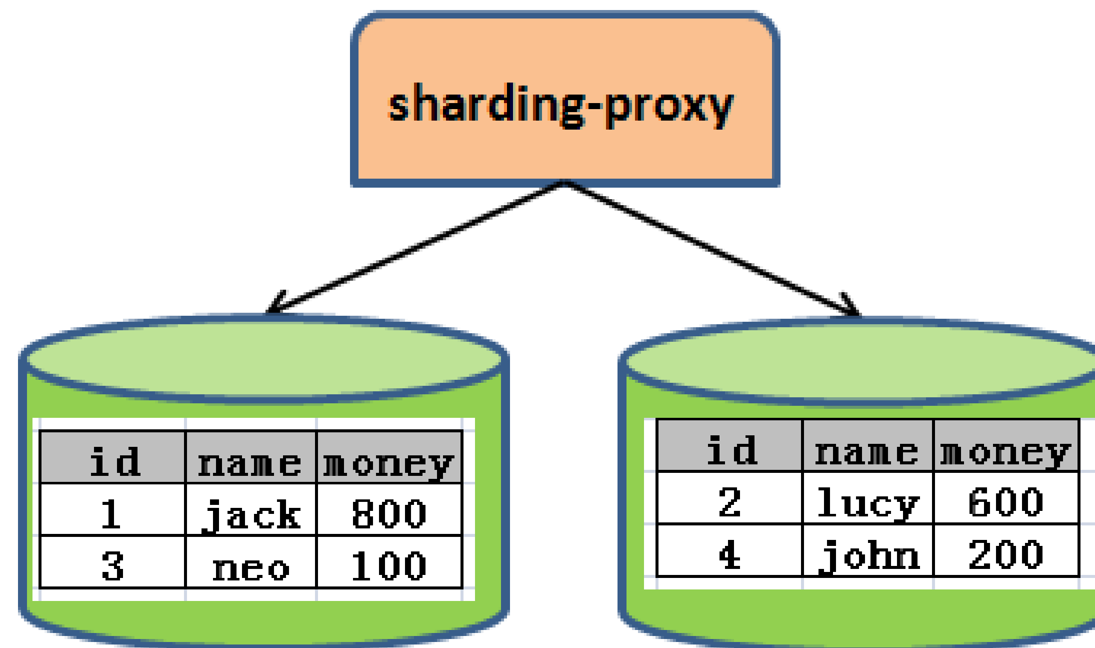
分布式事务问题的产生

UPDATE account SET money=money+500 WHERE id <4;

显式分布式事务：转账

```
BEGIN;
UPDATE account SET money=money-200 WHERE id =1;
UPDATE account SET money=money+200 WHERE id =2;
COMMIT;
```



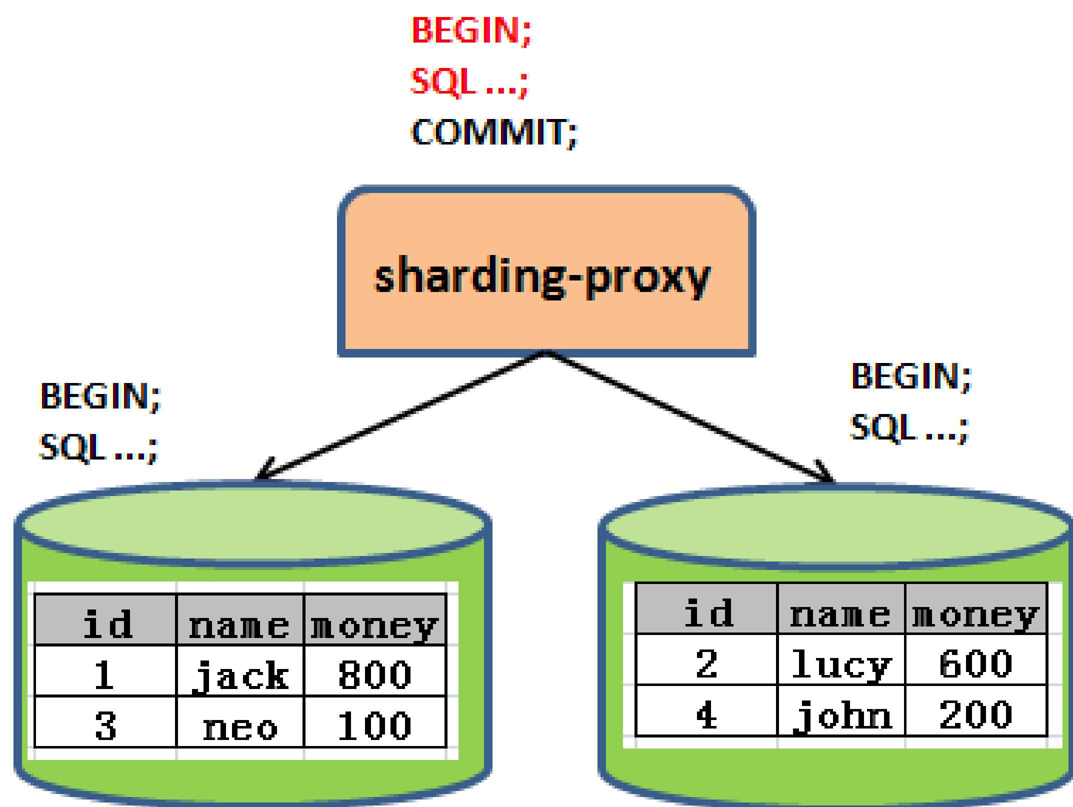


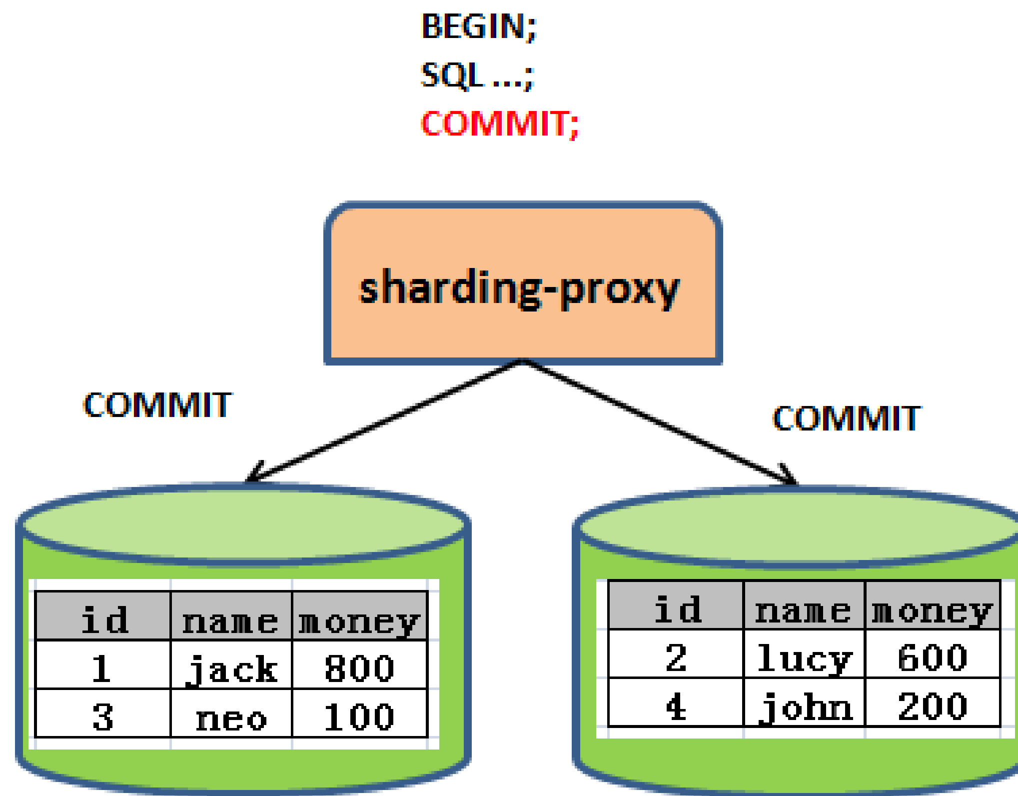
隐式分布式事务：发午餐补助

ACTION 爱可生 IT大咖说

分布式事务的实现-用普通事务简单实现

第一阶段：执行



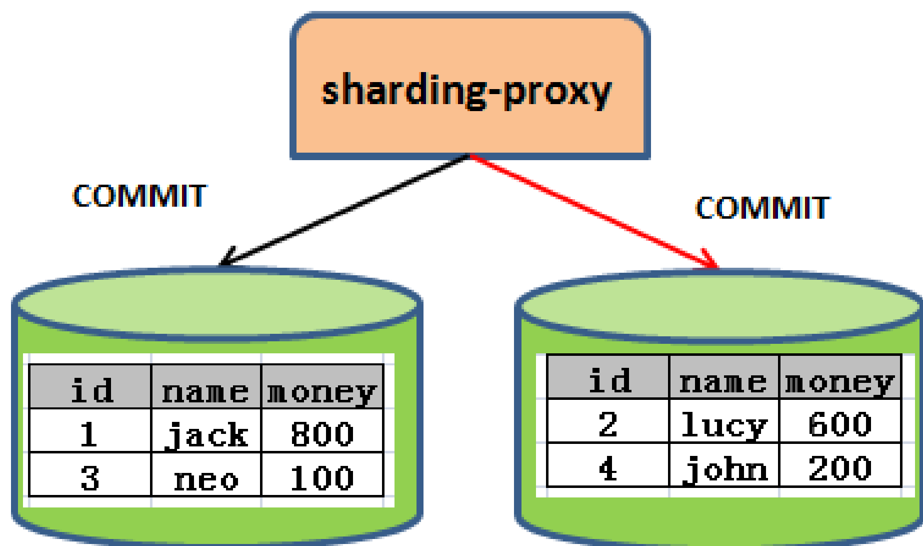


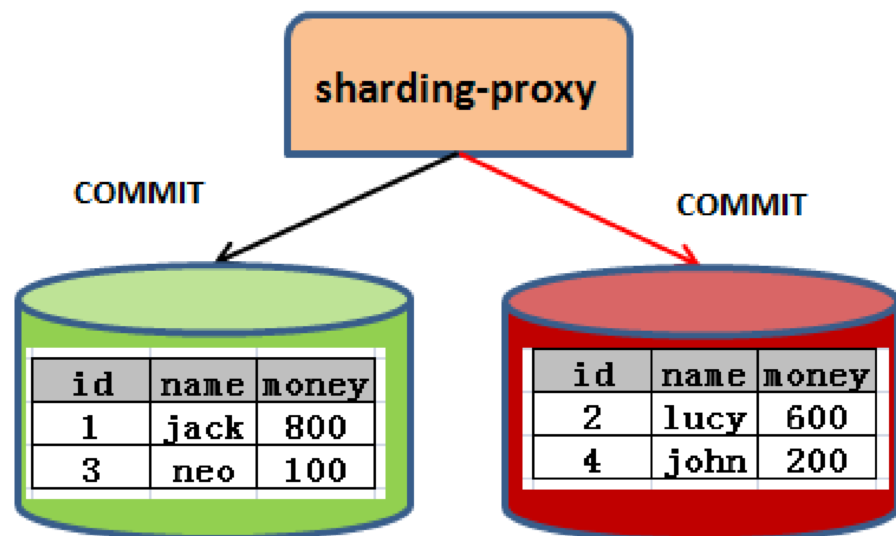
第二阶段：提交

ACTION 爱可生 TECHNOLOGY
IT大咖说 知识分享平台

分布式事务的实现-普通事务实现方式的问题

ACTION 爱可生 | IT大咖说 知识分享平台



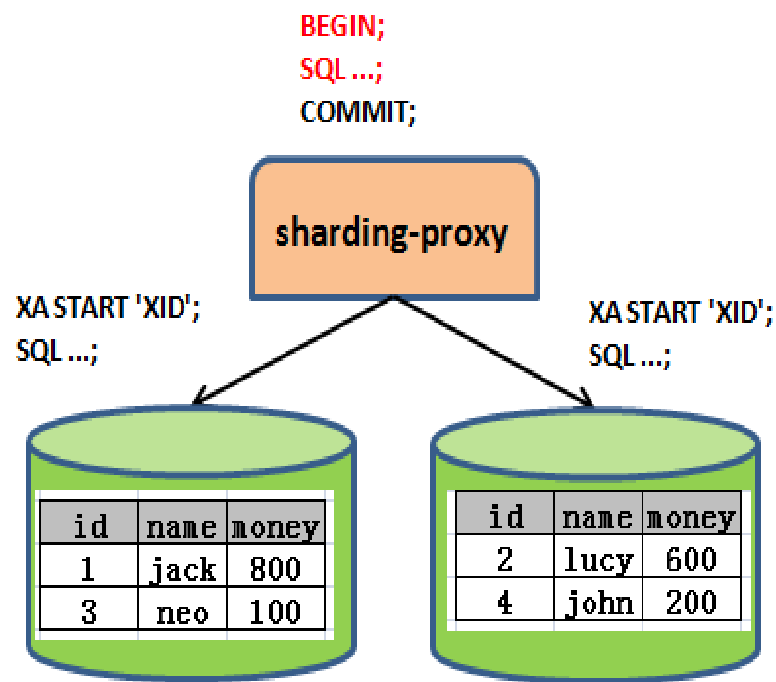


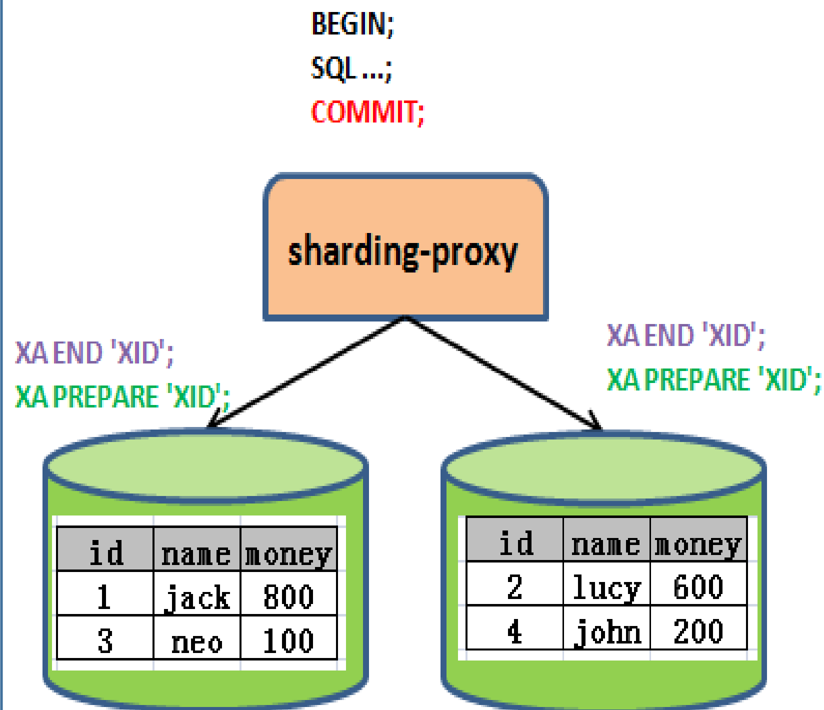
导致数据不一致可能的场景

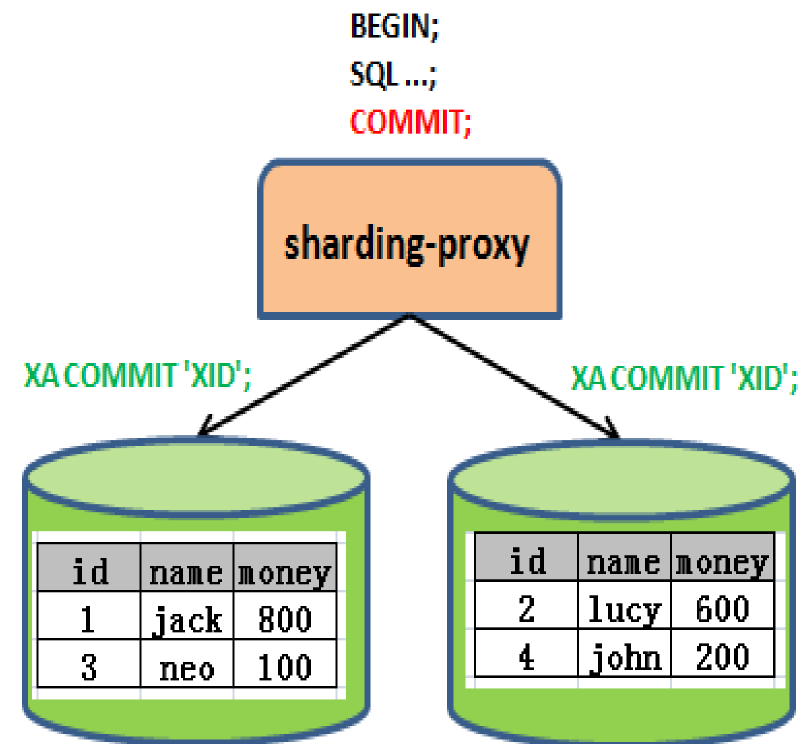
- 网络闪断
- 参与者数据库服务故障
- 参与者数据库宿主机宕机
- 协调者故障
- 协调者宿主机宕机

ACTION 爱可生

分布式事务的实现-MySQL XA事务







ACTION 爱可生 IT大咖说 知识分享平台

ACTION 爱可生

分布式事务的实现- MySQL XA事务

BEGIN;
SQL ...;
COMMIT;

sharding-proxy

XA COMMIT 'XID';

XA COMMIT 'XID';

id	name	money
1	jack	800
3	neo	100

id	name	money
2	lucy	600
4	john	200

- **网络闪断：**

已经完成PREPARE的XA事务，即使连接断开，事务状态也不会丢失。可以新建连接继续提交

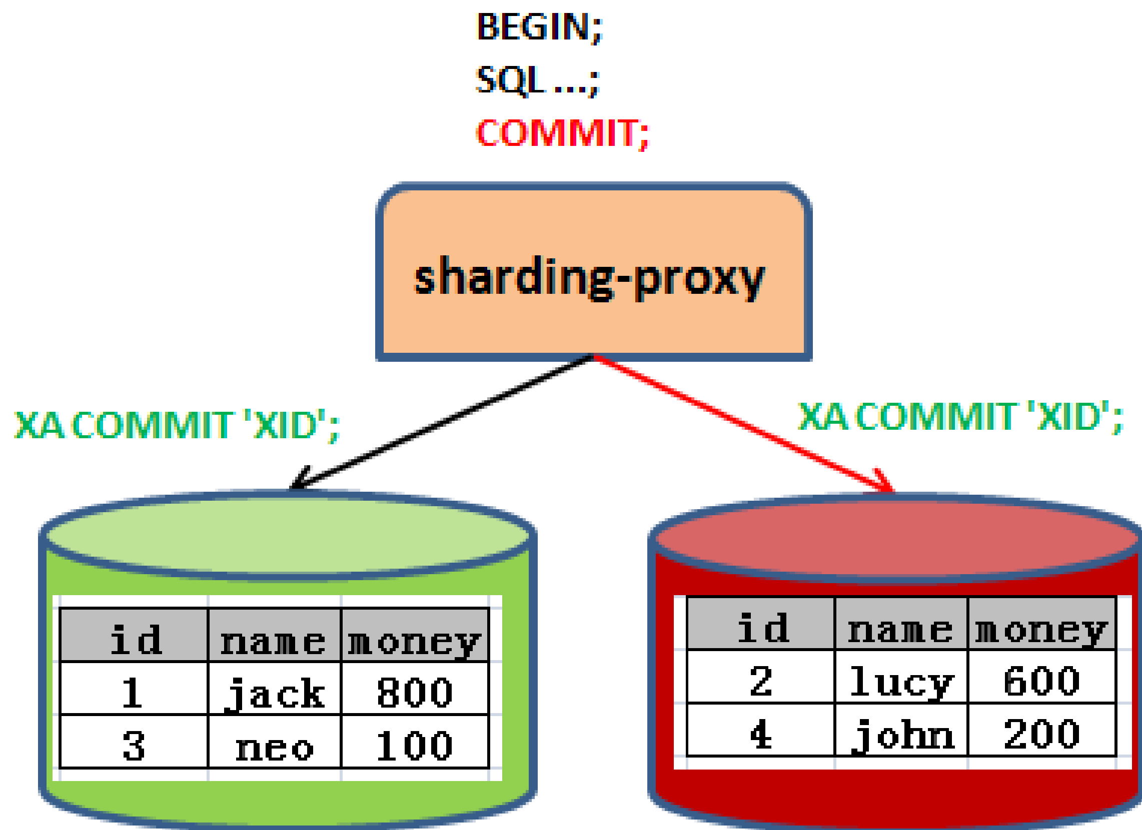
中间件可以设定次数阈值，多次执行，直到成功。

ACTION 爱可生 | IT大咖说 知识分享平台

ACTION 爱可生

分布式事务的实现- MySQL XA事务

ACTION 爱可生 | IT大咖说 知识分享平台



•参与者数据库服务故障

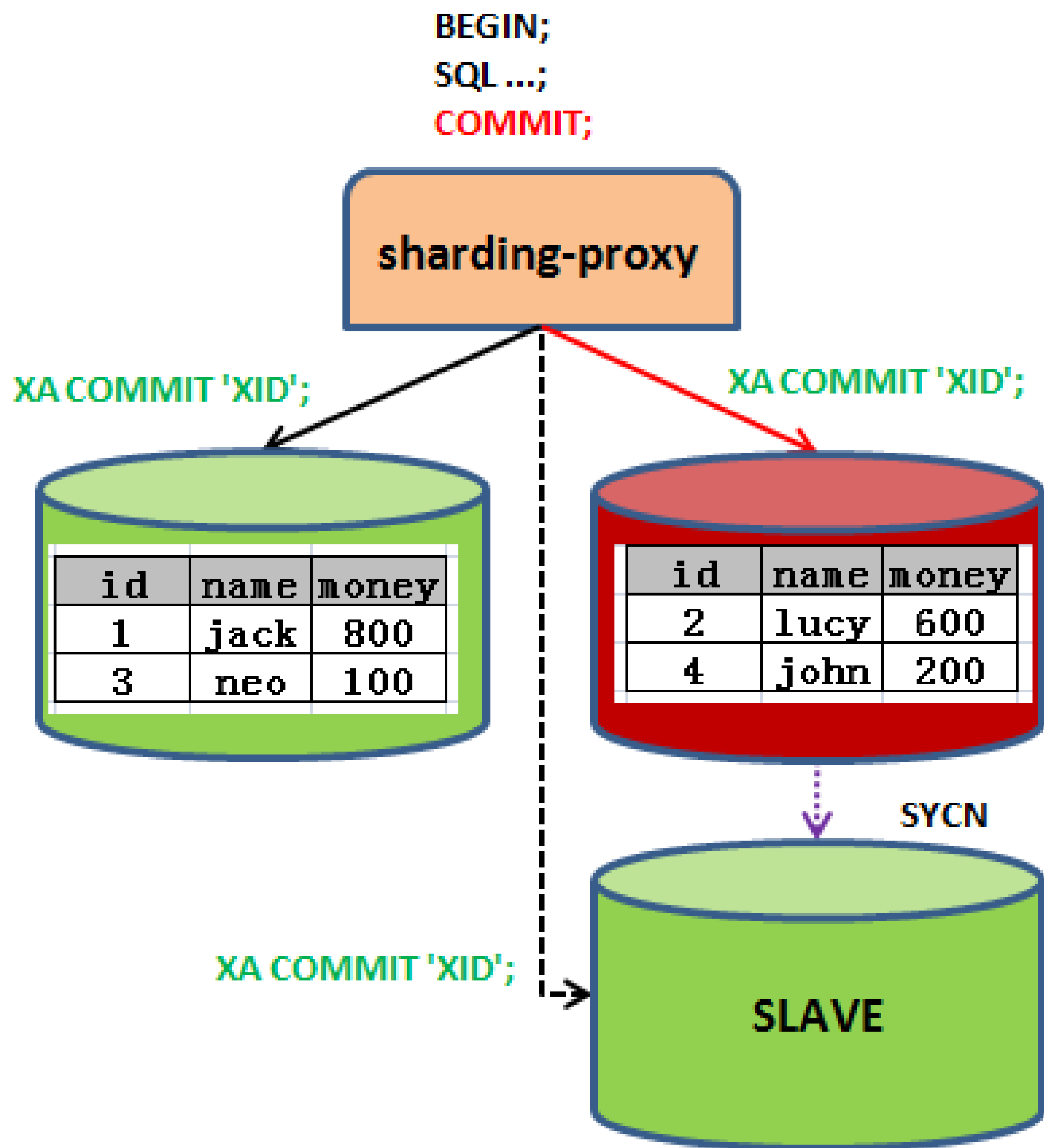
MySQL 5.7的 PREPARE状态会写入binlog

即使数据库服务重启， PREPARE状态也不会丢失。

中间件可以在数据库服务恢复后，继续提交。

ACTION 爱可生

分布式事务的实现- MySQL XA事务



- **参与者数据库宿主机宕机**

由于MySQL 5.7的 PREPARE状态会进binlog，如果数据结点采用了半同步等高可用方式，XA事务的PREPARE状态会同步到slave机器上.
即使发生了主从切换，状态也仍然不丢失

中间件可以在高可用切换之后，继续提交。

分布式事务的实现- MySQL XA事务

ACTION 爱可生 | IT大咖说 知识分享平台

协调者进程级故障：

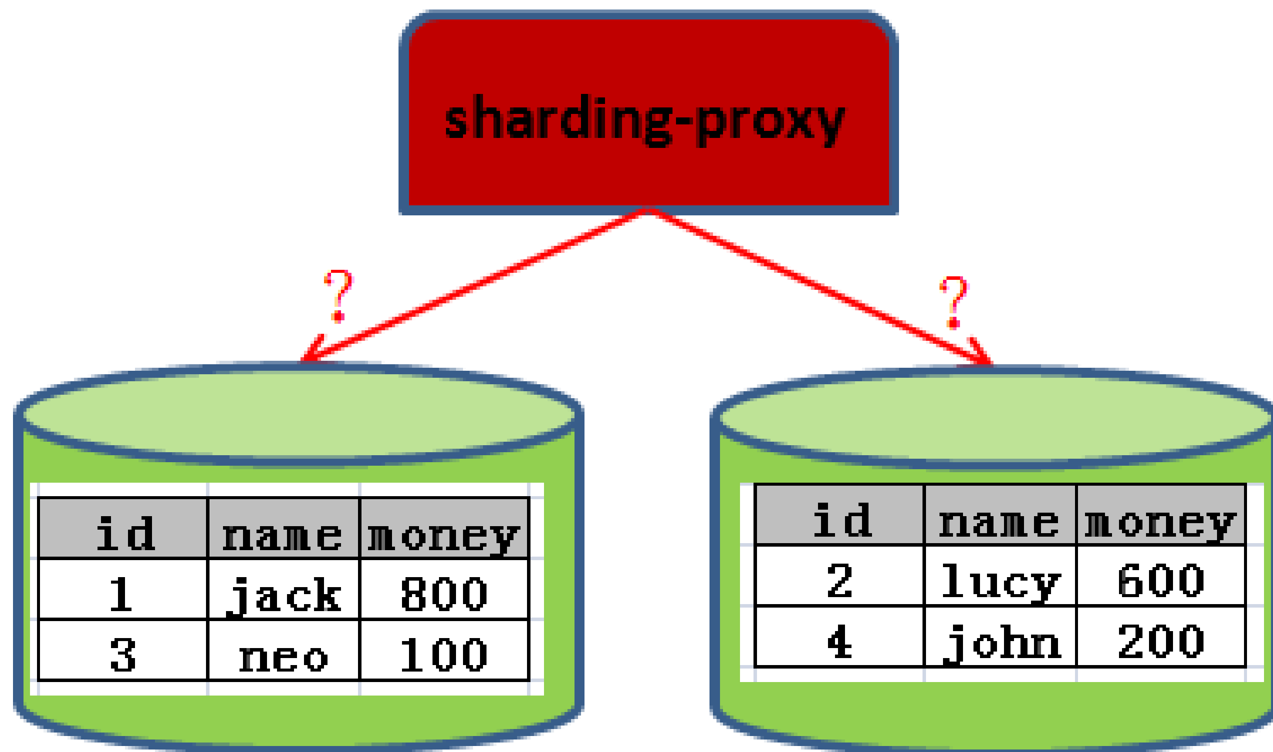
观察结点状态如下：

NODE1：

```
>xa recover;
Empty set (0.00 sec)
```

NODE2:

```
>xa recover;
+----------+--------------+--------------+------+
| formatID | gtrid_length | bqual_length | data |
+----------+--------------+--------------+------+
|        1 |            3 |            0 | xid  |
+----------+--------------+--------------+------+
```



id	name	money
1	jack	800
3	neo	100

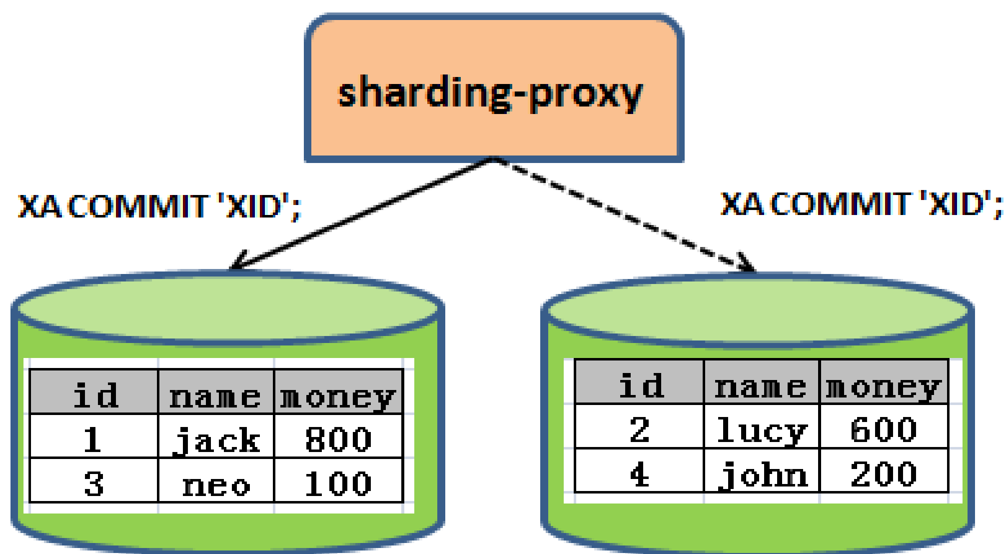
id	name	money
2	lucy	600
4	john	200

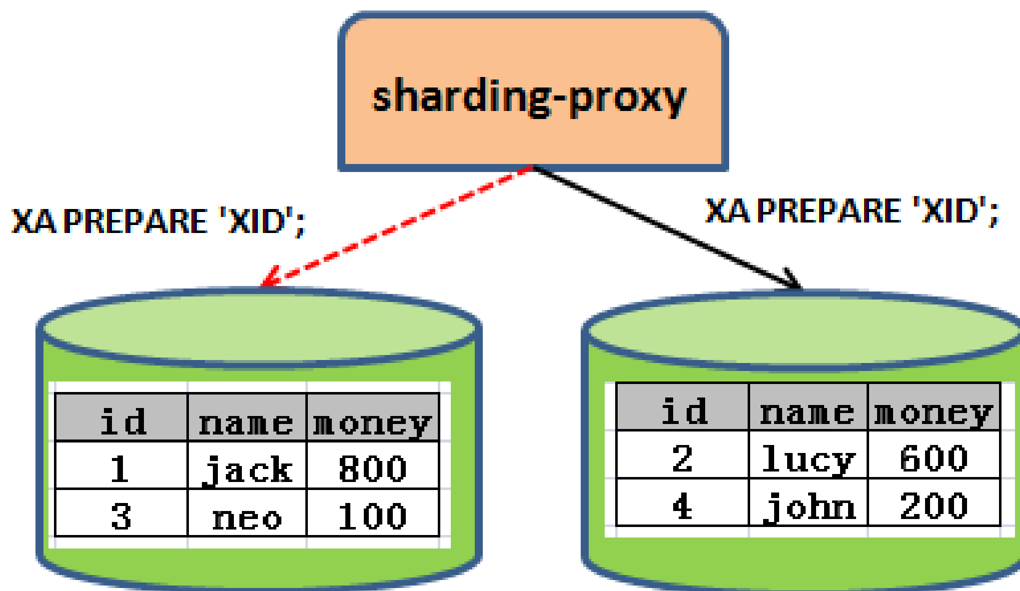
能否判断NODE 2上的事务该提交or回滚？

ACTION 爱可生

分布式事务的实现- MySQL XA事务 (协调者进程级故障)

无法判断node2上的事务该提交or回滚!





ACTION 爱可生 IT大咖说

分布式事务的实现- MySQL XA事务

协调者进程级故障：

解决方案：

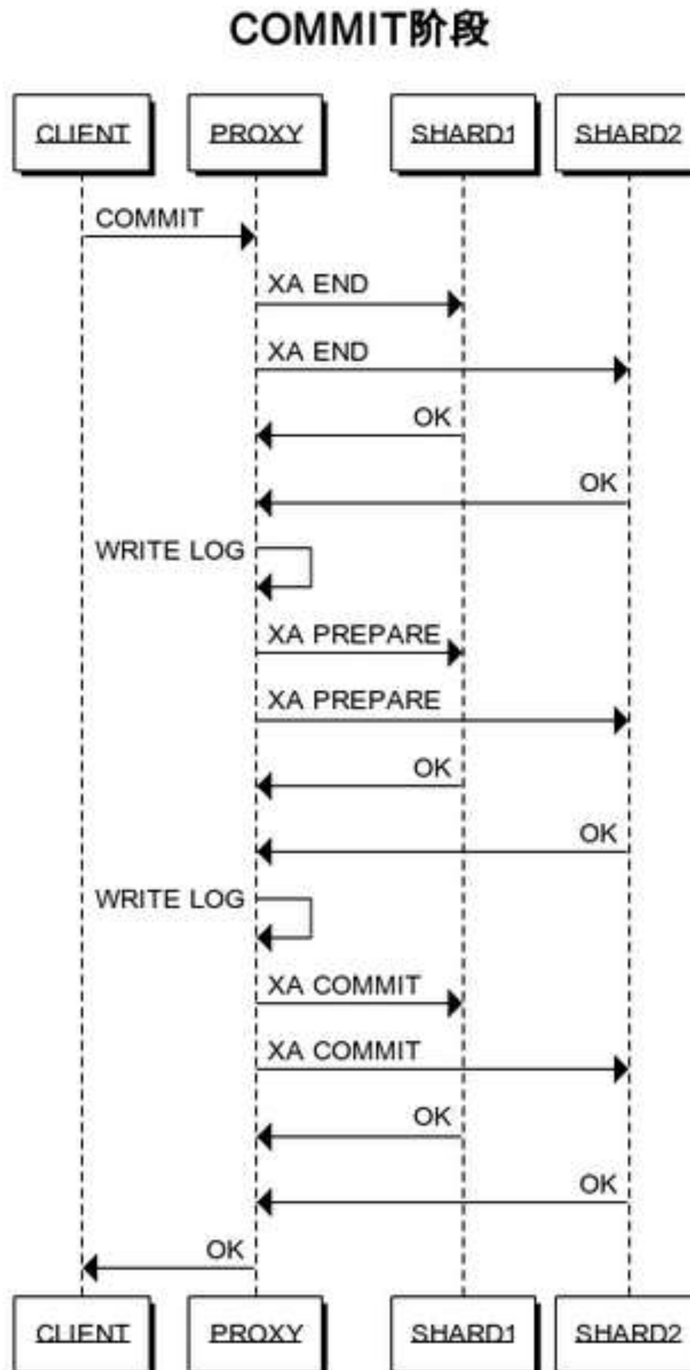
日志先行。

在XA PREPARE 和
XA COMMIT 之前
先将状态日志落盘。

异常发生后，可以根据日志进行手动恢复
或者自动恢复。

衍生问题：

写日志可能成为性能瓶颈



ACTION 爱可生 | IT大咖说 知识分享平台
ACTION 爱可生

分布式事务的实现- MySQL XA事务

协调者写日志性能问题:

启发：

MySQL Group Commit：

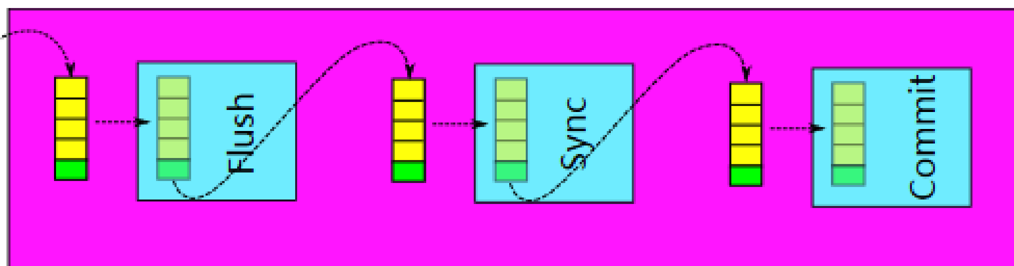
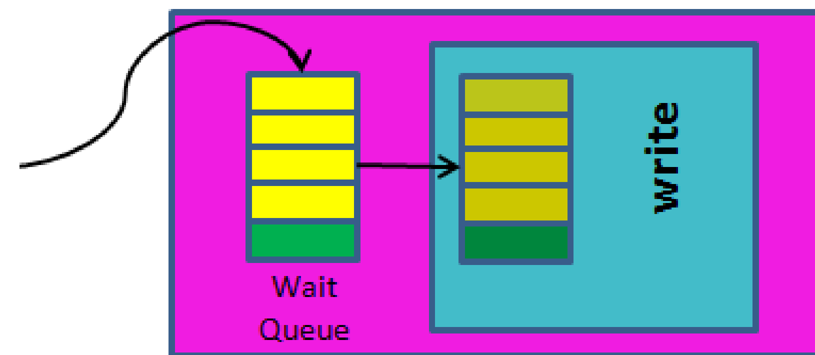


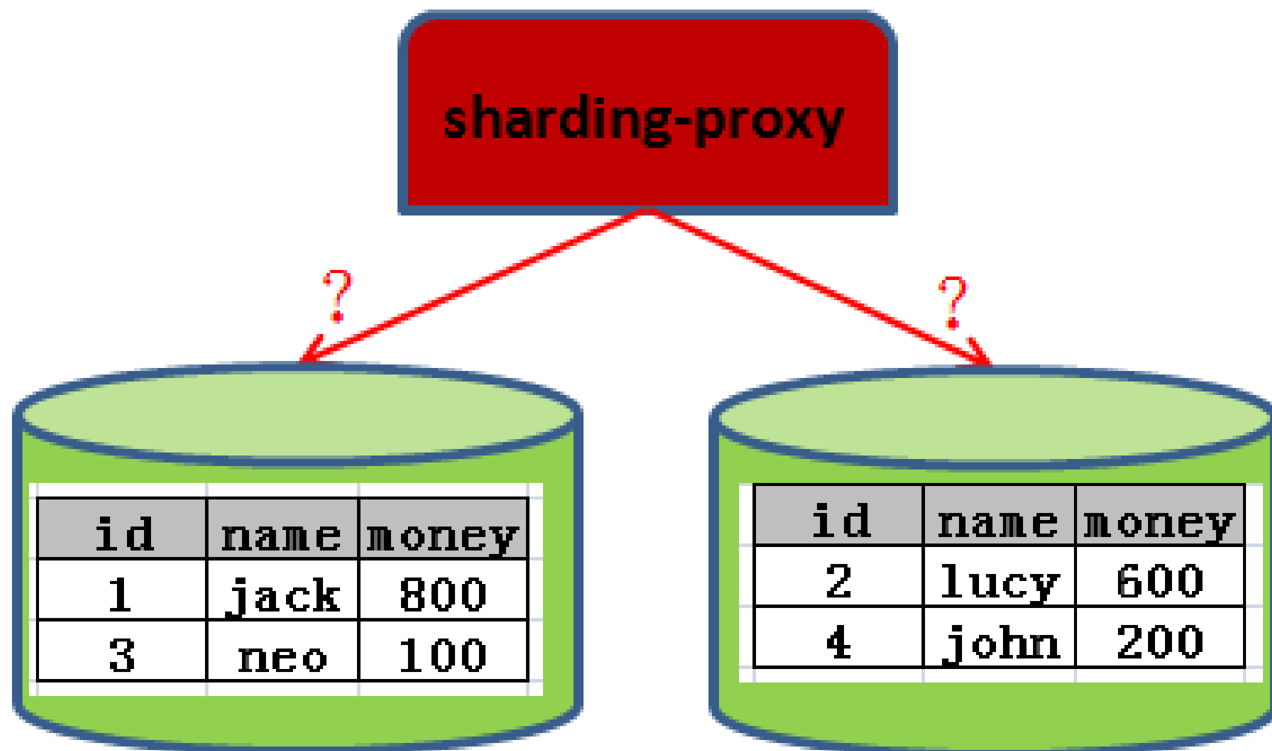
Figure 5. Commit Procedure Stages

思路 ：

dble Group Commit：



分布式事务的实现- MySQL XA事务



- **协调者宿主机故障：**

需要状态日志高可用：（Road Map）

- 写入ZooKeeper,Consul 等

ACTION 爱可生 | IT大咖说 知识分享平台

MySQL XA 事务实现分布式事务复盘

优点：

大幅度提高了分布式事务的健壮性。
在发生一些常见故障时候，能够做到数据的最终一致性。

缺点：

一定程度上降低了性能（写日志，多次网络交互等）。

限制：

需要工程实践上的外部依赖。
XA PREPARE状态不丢失 -> MySQL5.7 XA实现,MySQL配置。
数据结点服务器故障-> 数据节点的MySQL高可用
状态日志的不丢失->日志的高可用。

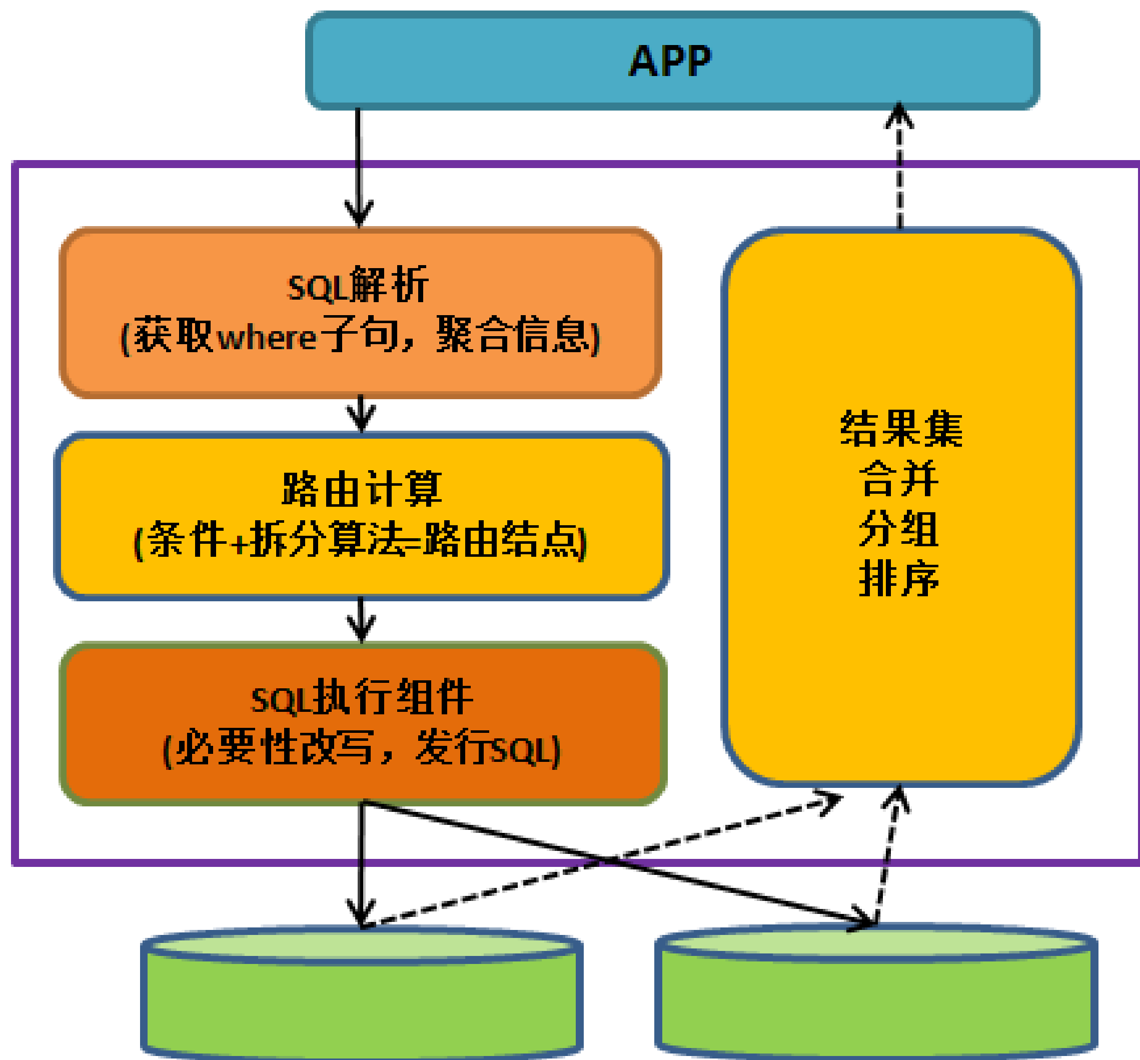
ACTION 爱可生

复杂查询实现

ACTION 爱可生 TECHNOLOGY
IT大咖说 知识分享平台

ACTION 爱可生 TECHNOLOGY

预备知识-一般中间件执行逻辑



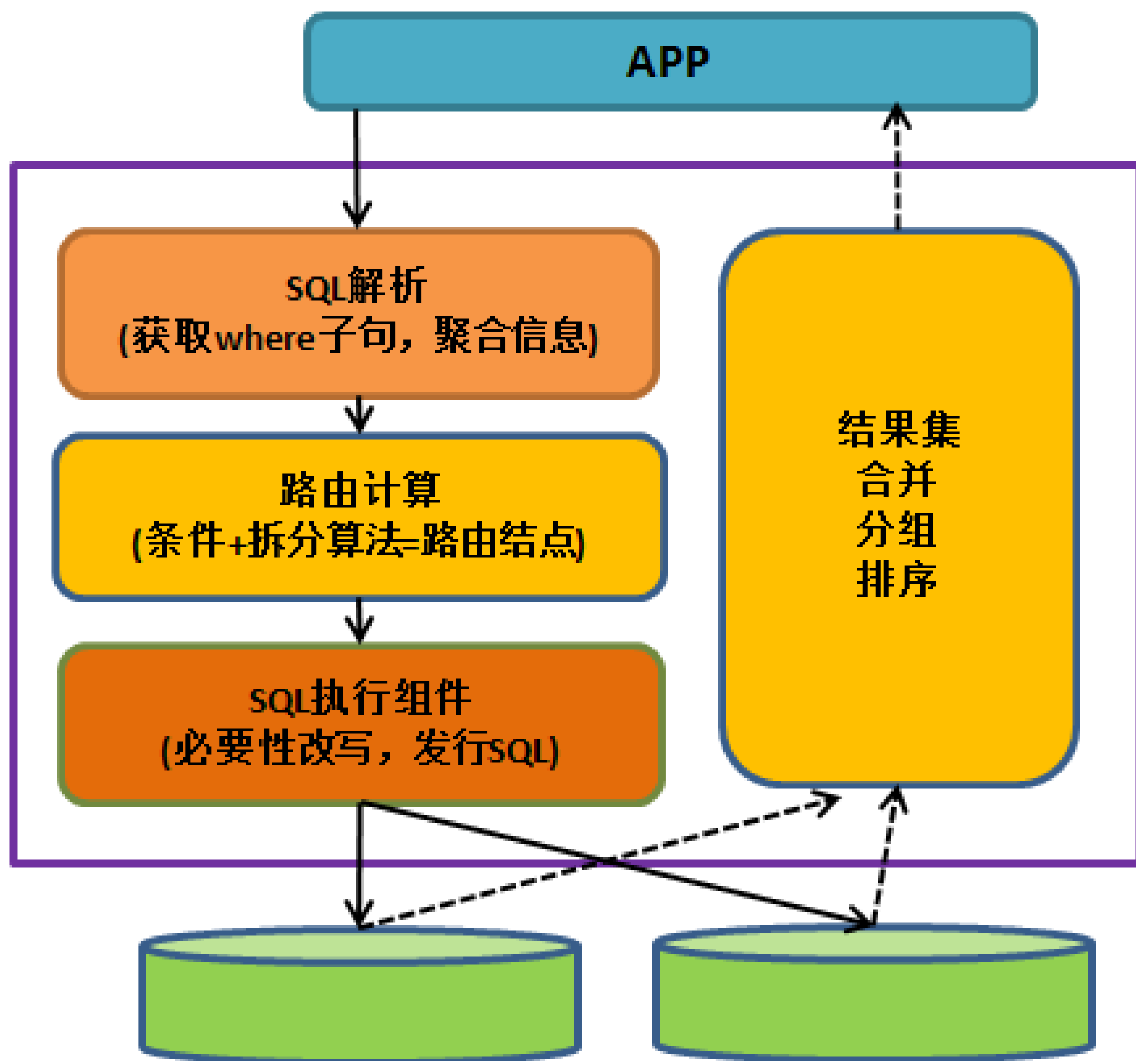
一般中间件内部执行逻辑

待解决问题：

- 复杂聚合
  例:select count(distinct column) …
- JOIN
- UNION
- Sub Query
- 以上混合复杂语句

预备知识-一般中间件执行逻辑

ACTION 爱可生 | IT大咖说 知识分享平台



工程上的解决方式：

- 复杂聚合：修改业务逻辑避开
- JOIN :尽量使用ER 表， global 表
- UNION：修改业务逻辑避开
- Sub Query：修改业务逻辑避开

这些方法都不错，但在不同和场景下，
成本代价相对高昂
有没有通用的解决方式呢？

ACTION 爱可生

复杂查询实现-关系代数

复杂查询实现思路:**关系代数 查询树**

1.解析SQL时候，将SQL转为基本元组，以及对元组进行关系运算，构建查询树 。

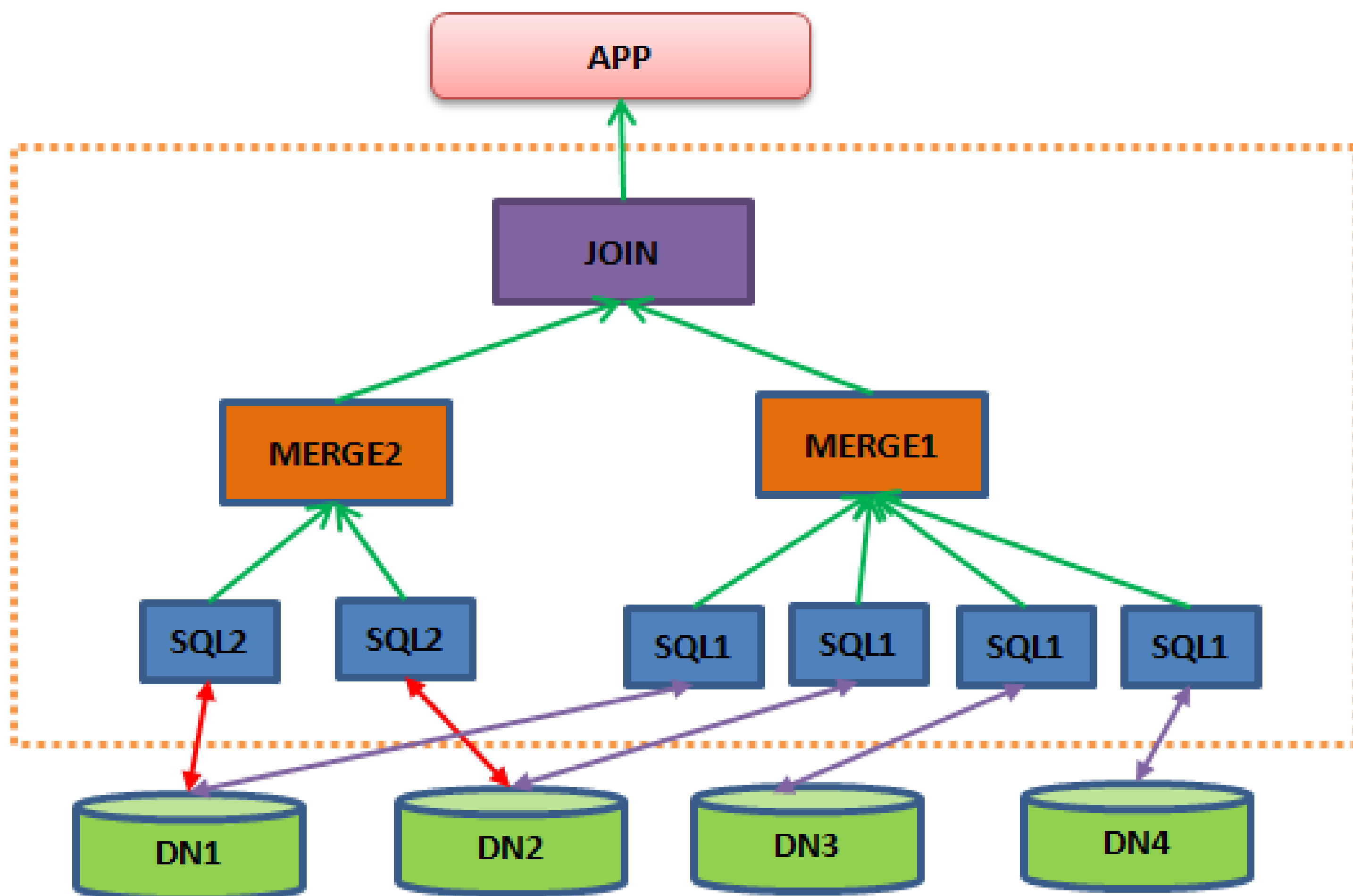
2.将基本元组及可下发的运算作为查询树的叶子节点，下发到物理数据结点查询。

3.结果集返回后，不可下发的运算作为查询树的非叶子节点对子树返回结果处理。

关系代数（部分）

Name	Symbol	对应sql部分
Selection	$\sigma_{a\theta b}(R)$ or $\sigma_{a\theta v}(R)$ θ包含⩽, ⩾, ≠, <, >, =	WHERE
Projection	$\Pi_{a_1,\ldots,a_n}(R)$	SELECT
Cartesian product	$A \times B = \{(a,b) \mid a \in A \text{ and } b \in B\}.$	笛卡儿积
set Union	$A \cup B = \{x : x \in A \text{ or } x \in B\}$	UNION
Rename	$\rho_{a/b}(R) = \{x \in B \mid x \notin A\}.$	别名
Natural join	$R \bowtie S = \{r \cup s \mid r \in R \land s \in S \land Fun(r \cup s)\}$	Natural join
θ-join	$R \bowtie_\theta S = \sigma\theta(R \times S)$	JOIN
equijoin	θ为"="的θ-join	
Semijoin	$R \ltimes S = \pi_{a1,\ldots,an}(R \bowtie S)$	
Left outer join	$R ⟕ S = (R \bowtie S) \cup ((R - \pi_{r_1,r_2,\ldots,r_n}(R \bowtie S)) \times \{(\omega, \ldots \omega)\})$	
Right outer join	$R ⟖ S = (R \bowtie S) \cup (\{(\omega, \ldots, \omega)\} \times (S - \pi_{s_1,s_2,\ldots,s_n}(R \bowtie S)))$	
Full outer join	$R ⟗ S = R ⟗ S = (R ⟕ S) \cup (R ⟖ S)$	
Aggregation	Exp1,Exp2,Exp3...Gfunc1,func2,func3...	聚合

复杂查询实现-JOIN



JOIN 举例：

select * from table1 a inner join table2 b on a.id =b.id ;

table1 有四个分片dn1~4

table2 有2个分片 dn1,dn2

查询树：

下发语句：

SQL1:select * from table1;

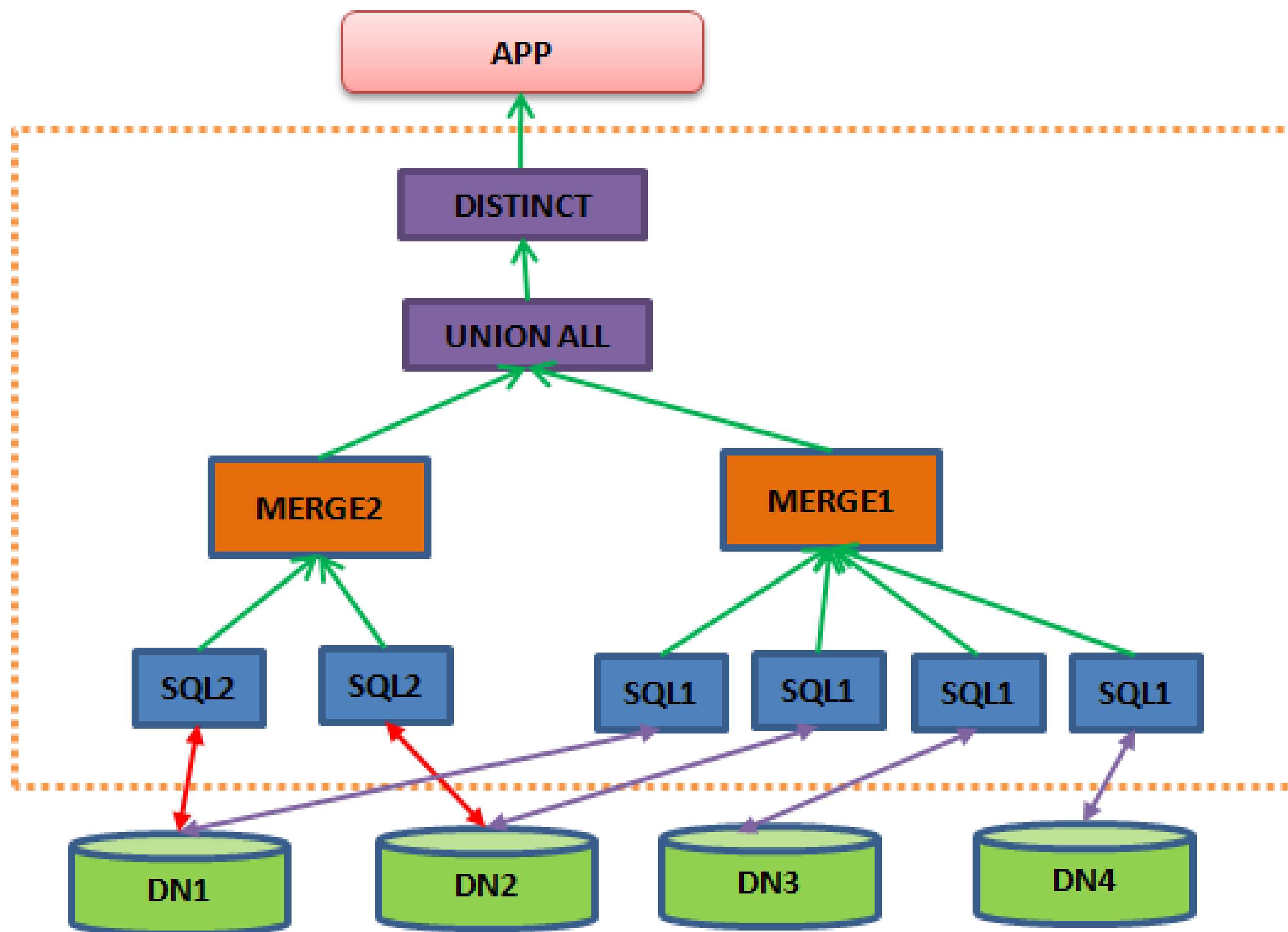
SQL2:select * from table2;

中间件后续操作：

1.各个节点结果合并

2.将结果集按照 a.id =b.id过滤及JOIN

复杂查询实现-UNION



UNION 举例：

select id,name from table1 a union select id,name from table2 b ;

table1 有四个分片dn1~4

table2 有2个分片 dn1,dn2

查询树：

下发语句：

SQL1:select id,name from table1 ;

SQL2:select id,name from table2;

中间件后续操作：

1.各个节点结果合并

2.将结果集UNION All 操作

3.对结果集去重

复杂查询实现-复杂聚合

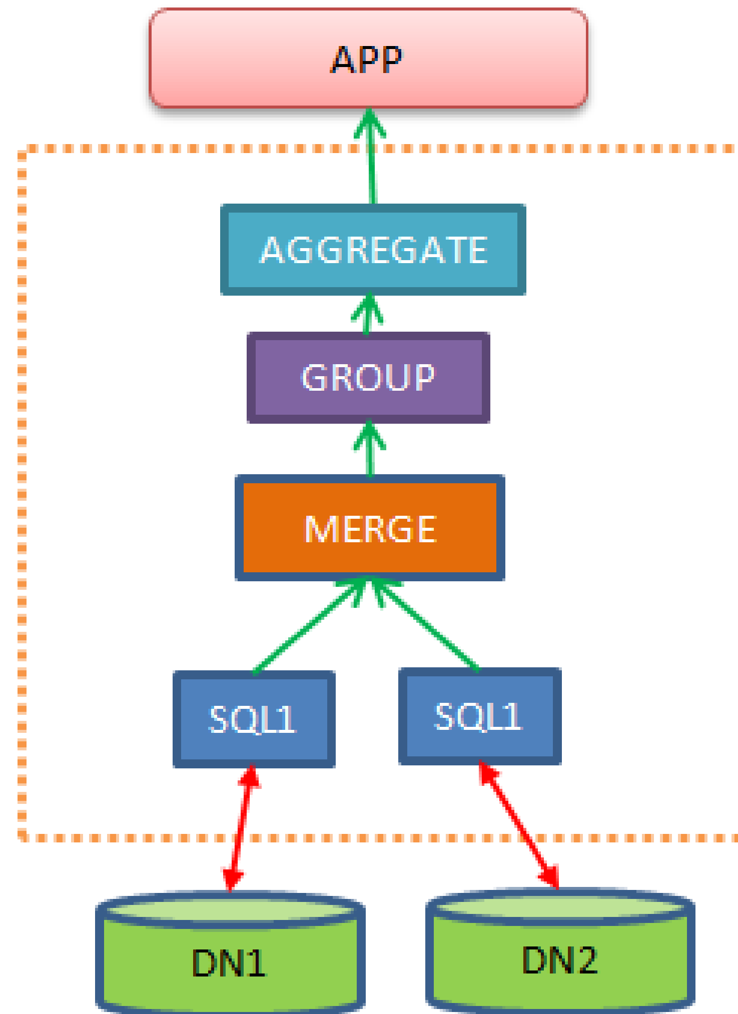
ACTION 爱可生 | IT大咖说 知识分享平台

聚合 举例：
select count(distinct col) from table2;
table2 有2个分片 dn1,dn2

查询树：

下发语句：
SQL1:select distinct col from table1 ;

中间件后续操作：
1.各个节点结果合并
2.对结果分组合并
3.计算聚合函数COUNT



ACTION 爱可生

复杂查询实现-子查询

子查询举例：

select * from table1 a where id> (select id from table2 b where name ='test') ;

table1 有四个分片dn1~4

table2 有2个分片 dn1,dn2

查询树1：

下发语句：

SQL1:s select id from table2 b where name ='test';

中间件后续操作：

1.各个节点结果合并

2.对结果进行判断生成New Query，二次查询树或者报错给客户端

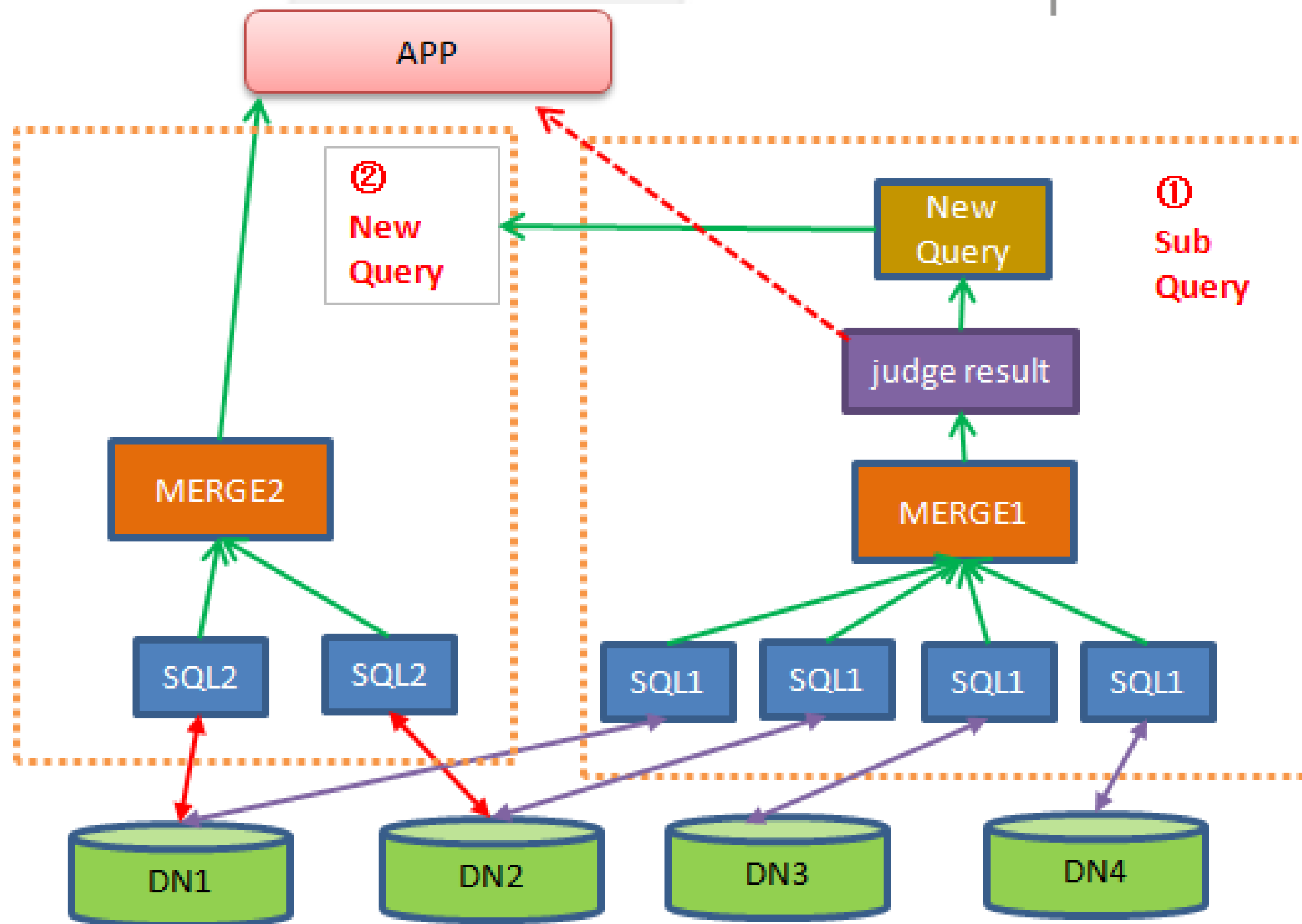
查询树2：

下发语句：

select * from table1 a where *condition*

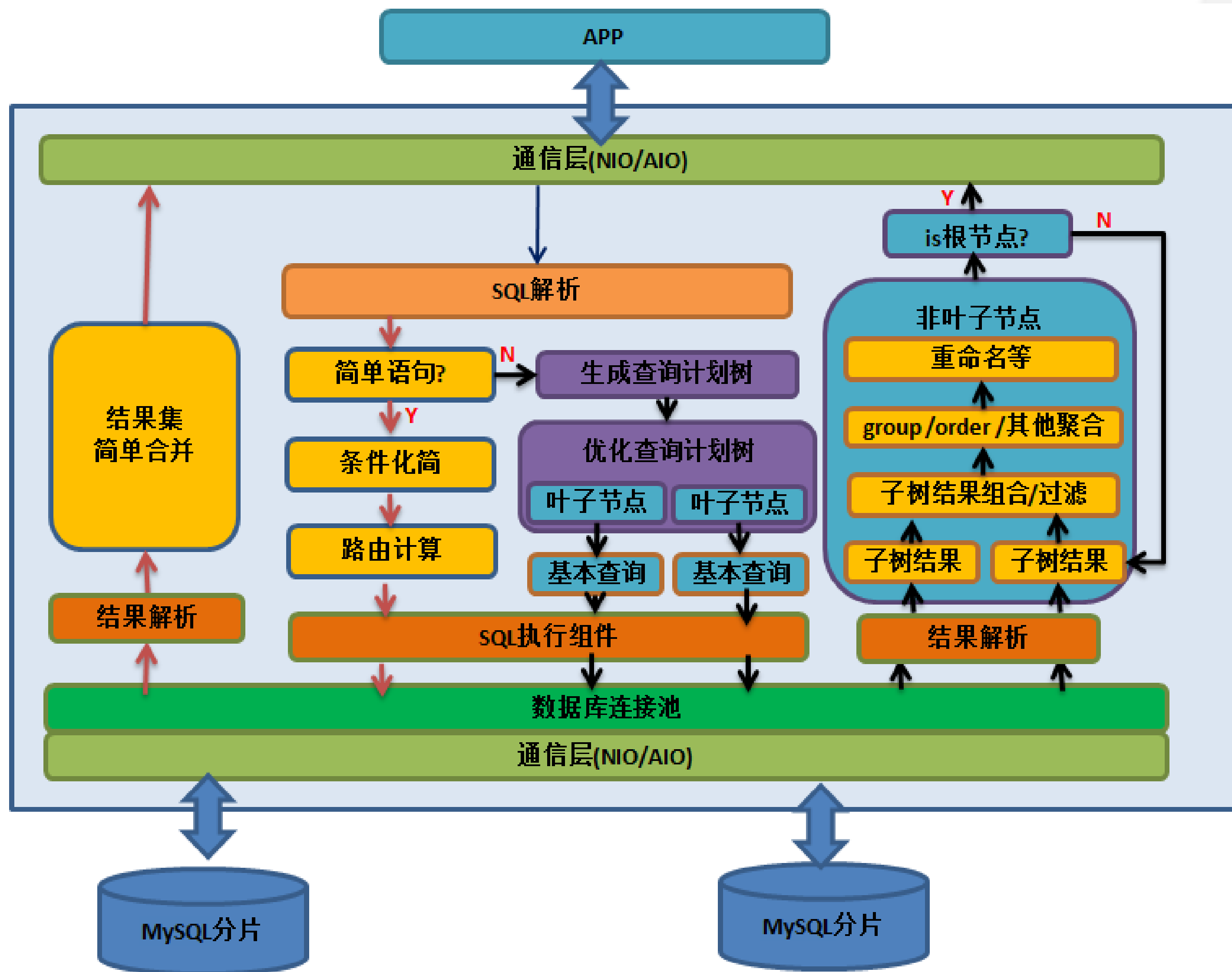
中间件后续操作：

各个节点结果聚合



ACTION 爱可生 TECHNOLOGY
IT大咖说 知识分享平台
ACTION 爱可生 TECHNOLOGY

dble 内部结构（执行逻辑）



通过构建查询树的方式
可以支持各种复杂查询

- 聚合分组
- JOIN
- UNION (ALL)
- Sub Query
- 复杂类型/表达式的排序
- 视图（on Road Map）
- 以上组合

查询优化举例

ACTION 爱可生 TECHNOLOGY
IT大咖说 知识分享平台

ACTION 爱可生 TECHNOLOGY

查询优化举例

ACTION 爱可生 | IT大咖说 知识分享平台

优化（改写下发SQL）目的：

- 正确的查询结果
- 生成良好的查询计划
- 优化查询速度
- 构建适应分布式场景的查询计划

ACTION 爱可生

查询优化举例

优化原理：

- 关系代数的等价规则
  - Relational Algebraic Equivalence Transformation Rule.pdf
- 聚合操作查询优化
  - [Klug]Equivalence of Relational Algebra and Relational Calculus Query Languages Having Aggregate Functions
  - [Yan and Larson]Eager Aggregation and Lazy Aggregation.pdf
  - [S Chaudhuri， K Shim]Including Group-By in Query Optimization .pdf
- 外连接查询优化
  - [Rosenthal and Reiner]Extending the Algebraic Framework of Query Processing to Handle Outerjoins.pdf
  - [Galindo-Legaria and Rosenthal]How to extend a conventional optimizer to handle one- and two-sided outer join.pdf
  - [Galindo-Legaria]outerjoins as disjunctions .PDF
- 关联子查询优化
  - [Won Kim]On Optimizing an SQL-like Nested Query.pdf
  - [Ganski and Wong]Optimization of Nested SQL Queries Revisited.pdf
  - [Galindo-Legaria and Joshi 2001]Orthogonal Optimization of Subqueries and Aggregation.pdf

ACTION 爱可生 | IT大咖说 知识分享平台

ACTION 爱可生

ACTION 爱可生 TECHNOLOGY | IT大咖说 知识分享平台

分布式场景下的一些优化经验：

- 正确性为第一要务
- 利用已有的全局/ER关系表
- 减少中间件与数据库之间的数据传输
- 尽量将能下发的计算下发给结点完成
- 减少中间件运算的空间/时间复杂度

ACTION 爱可生 TECHNOLOGY

ACTION 爱可生 | IT大咖说 知识分享平台

规则:保证LIMIT正确性

例 ：

原始语句：select * from tb_test order by id limit 20,10；

如果原样下发到各个结点，则返回的结果集再合并最终结果不能保证正确

优化后下发的语句:select * from tb_test order by id limit 0,30(20+10).

ACTION 爱可生

查询优化举例

ACTION 爱可生 | IT大咖说 知识分享平台

规则： JOIN 查询树生成时将与JOIN条件相关列添加排序

例 ：

select * from table1 a inner join table2 b on a.id =b.id ；

下发语句：

SQL1:select * from table1;

SQL2:select * from table2;

中间件时间复杂度:

设SQL1返回M行，SQL2返回N 行，比较运算为 O(M*N).

改写为：

SQL1:select * from table1 order by id;（p个分片）

SQL2:select * from table2 order by id;（q个分片）

增加了节点排序的成本，相当于多路并发排序

中间件比较运算时间复杂度:

合并结点数据复杂度A ：2个多路有序归并排序=O(p*logM)+O(q*logN)

Join时间复杂度B ：最小O(Min(M,N)),最大O(M+N-1).

总复杂度Z=A+B.

可以流水线处理

ACTION 爱可生

查询优化举例

ACTION 爱可生 TECHNOLOGY | IT大咖说 知识分享平台

规则: Having中与聚合无关的条件可以合并到where中

例 ：

select id from test group by id having col2 ='test' and count(*)>2;

下发语句：

SQL1: select id from test;

需要拉取全量的test 表数据在中间件处理

改写为：

SQL:select test.id from test where test.col2 =‘test’；

TECHNOLOGY ACTION 爱可生

查询优化举例

规则:查询树父节点约束条件推向内部叶子节点

例 ：

select * from table1 inner join table2 on table1.name =table2.name where table1.id>5 and table2.id<10;

下发语句：

SQL1:select * from table1 order by name;
SQL2:select * from table2 order by name;
需要拉取全量的table2表数据在中间件处理

改写为：

SQL1：select * from table1 where table1.id>5 order by name
SQL2：select * from table2 where table2.id<10 order by name
条件下推，减少从结点获取的数据量

ACTION 爱可生 TECHNOLOGY | IT大咖说 知识分享平台

规则:NEST LOOP JOIN（需配置启用）

例 ：

select * from table1 ,table2 where table1.id=table2.id and table1.name='test';

下发语句：

SQL1:select * from table1 where table1.name='test' order by id;
SQL2:select * from table2 order by id;
需要拉取全量的table2表数据在中间件处理

改写为：

SQL1:select * from table1 where table1.name='test' order by id;（驱动表）
SQL1:select * from table2 where id in(sql1 的id list)

注：目前驱动表的选择还很粗糙，未来期望1.通过维护数据统计直方图，2.hint来解决。

ACTION 爱可生 TECHNOLOGY

查询优化举例

规则:利用global和ER关系表

1. global 表优化:多表查询尽量与global表结合在一起下发。
   例如配置：

```
<table name="sharding_two_node" primaryKey="id" dataNode="dn1,dn2" rule="two_node_hash"/>
<table name="global_two_node" primaryKey="id" dataNode="dn1,dn2" type="global"/>
```

SQL: select * from sharding_two_node a inner join global_two_node b on a.id =b.id;
此句SQL将会作为查询树的叶子节点整体下发

2. 父子表/ER表优化:符合ER查询的语句尽量结合在一起下发。
   例如配置：

```
<table name="er_parent" primaryKey="ID" dataNode="dn1,dn2,dn3,dn4" rule="four_node_hash">
    <childTable name="er_child1" primaryKey="child1_id" joinKey="child1_id" parentKey="id"/>
</table>
```

SQL: select * from er_parent a inner join er_child1 b on a.id =b.child1_id；
此句SQL将会作为查询树的叶子节点整体下发

ACTION 爱可生 | IT大咖说 知识分享平台

感谢乐于分享的开源精神
感谢参与开源的贡献者们

良好的项目演化需要你的参与
无论是代码贡献，**bug**发现提出，文档修订
我们全都欢迎
开源地址:**https://github.com/actiontech/dble**

TECHNOLOGY ACTION 爱可生

THANK YOU

TECHNOLOGY
ACTION
爱可生