数据库平台化技术-MySQL

--刘小成　　2012/05/01

问题引入

数据量和服务器数量的增加，数据库层面临问题：

- 对应用程序透明化
- 高可用
- 负载均衡
- 并发控制

Replication

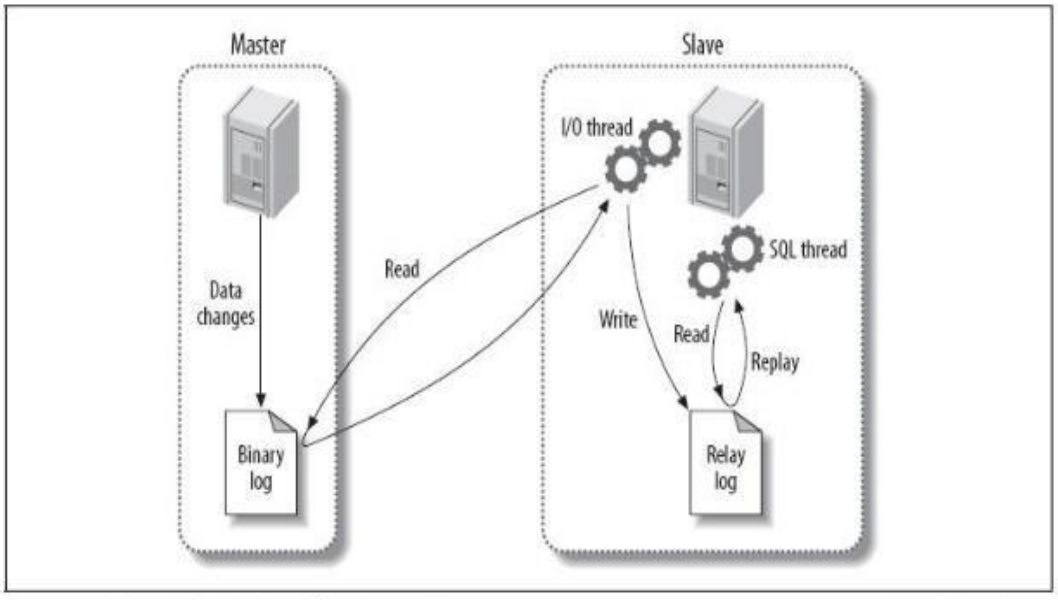
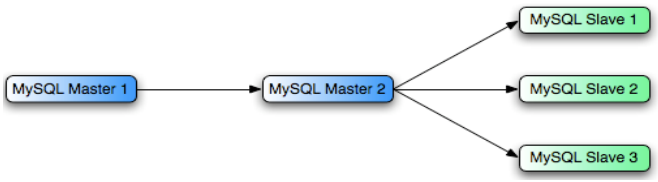


Figure 8-1. How MySQL replication works

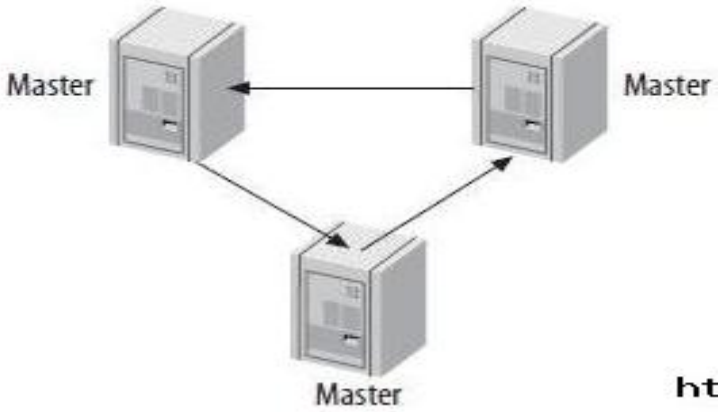
1. Master
   a) Binlog flush 方式。Sync_binlog（0、1、n）。
   b) Binlog 登记格式。Row/statement/mixed
2. Slave
   a) IO && SQL thread。
   b) 错误处理。
   c) 时延。

3. M-S

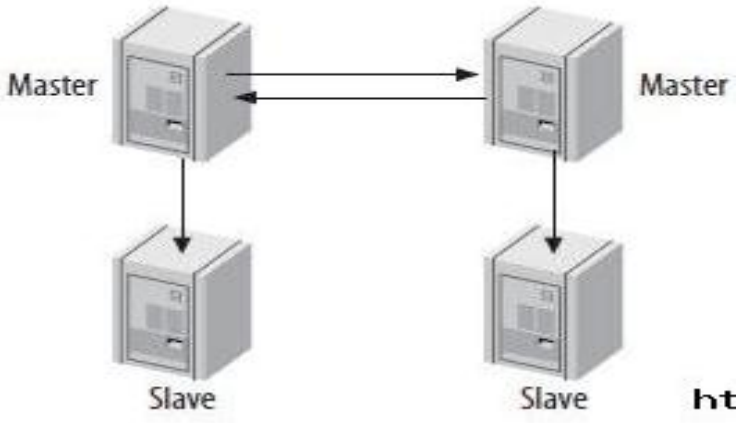
1:n



1:1:n



1:1:1



n:n

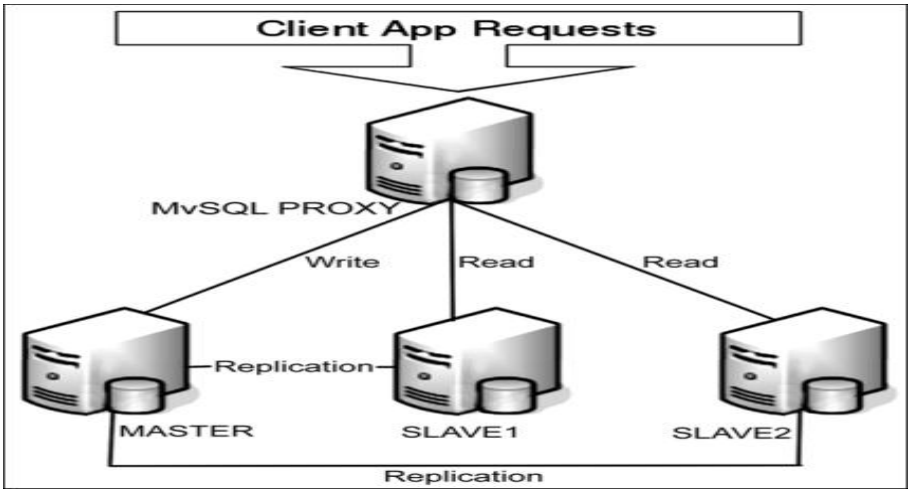
常用方案

程序端读写分离

通过在程序端配置多数据源，在数据库连接模块中分离读写 sql。实现数据的多写，读写分离等。 简化数据库结构，程序直接访问 db，但 db 的结构变化直接影响程序配置，扩展性低。

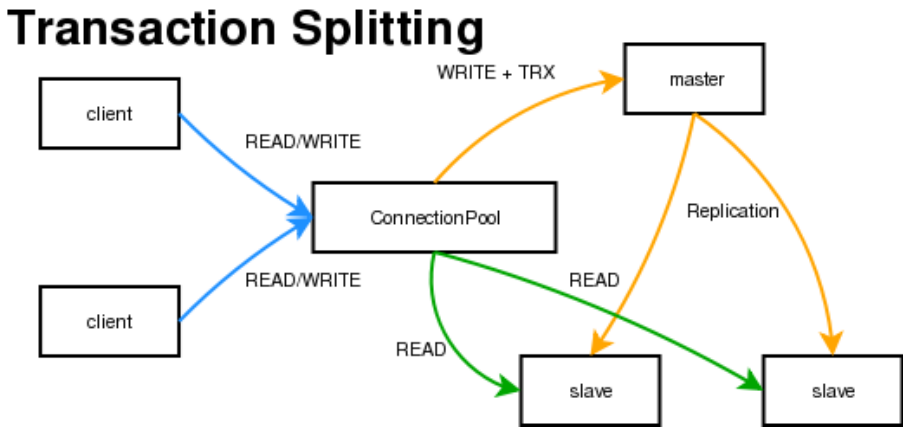
Proxy

a) Mysql proxy



Lua 配置灵活：连接服务器(connect_server)、连接握手(read_handshake)、提交验证(read_auth)、验证结果(read_auth_result)、提交查询(read_query)、查询结果(read_query_result)。稳定性一般，管理功能少，Alpha release（5.5.*）不适用于生产环境。

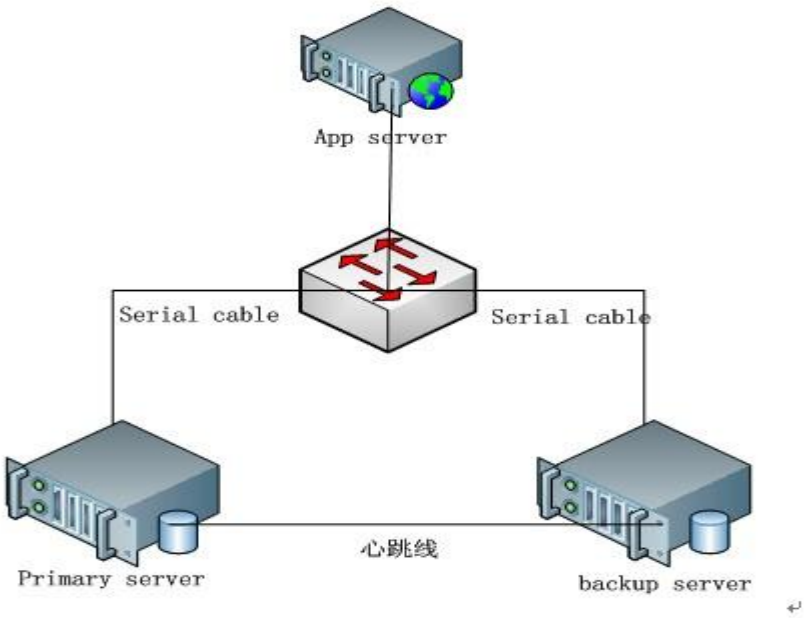
b) Amoeba for mysql



负载均衡、高可用性、sql 过滤、可承受高并发、Query Route、数据切分。不支持事务，ddl 语句只在默认 server 上执行，不支持跨节点 join、排序，不支持分库分表。

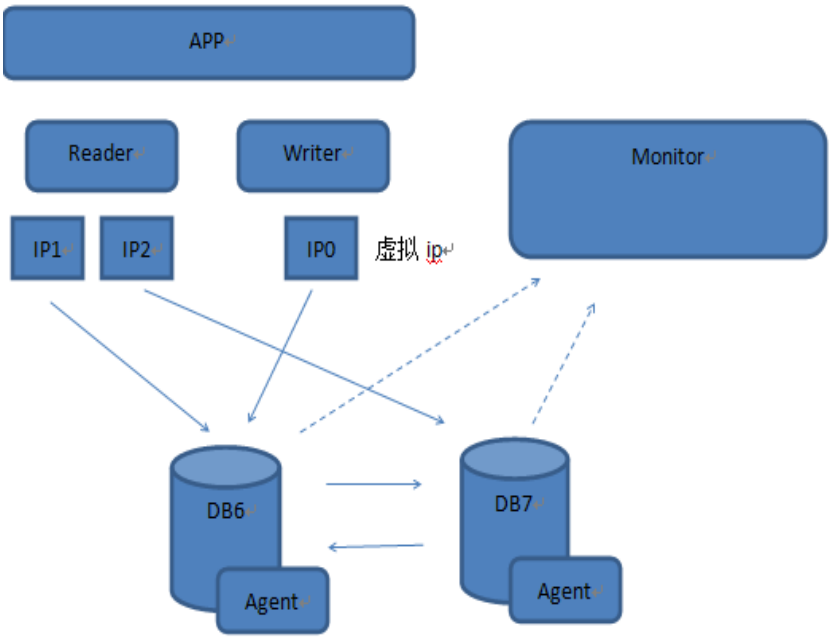
Ha

c) Linux Heartbeat



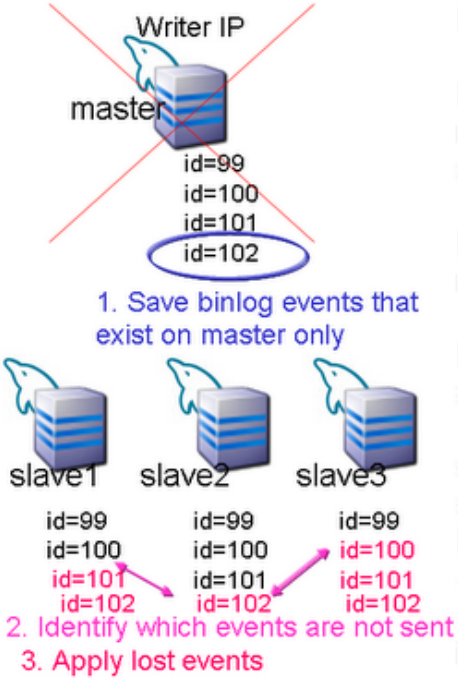
系统级别的 failover。

d) Mmm



Mysql 复制级别的 failover。包含主机监控、mysql 服务监控、复制线程监控等多方位监控手段，在 MySQL 多主环境中，使用它可以在任何一个节点（master 或 slave）出现故障时能够及时作出判断和调整。它使用虚拟 ip 技术使数据库架构内的角色切换对上层的应用程序透明。

e) Mha



MySQL replication is asynchronous.

It is likely that some (or none of) slaves have not received all binary log events from the crashed master.

It is also likely that only some slaves have received the latest events.

In the left example, id=102 is not replicated to any slave.

slave 2 is the latest between slaves, but slave 1 and slave 3 have lost some events.
It is necessary to do the following:
- Copy id=102 from master (if possible)
- Apply all differential events, otherwise data inconsistency happens.

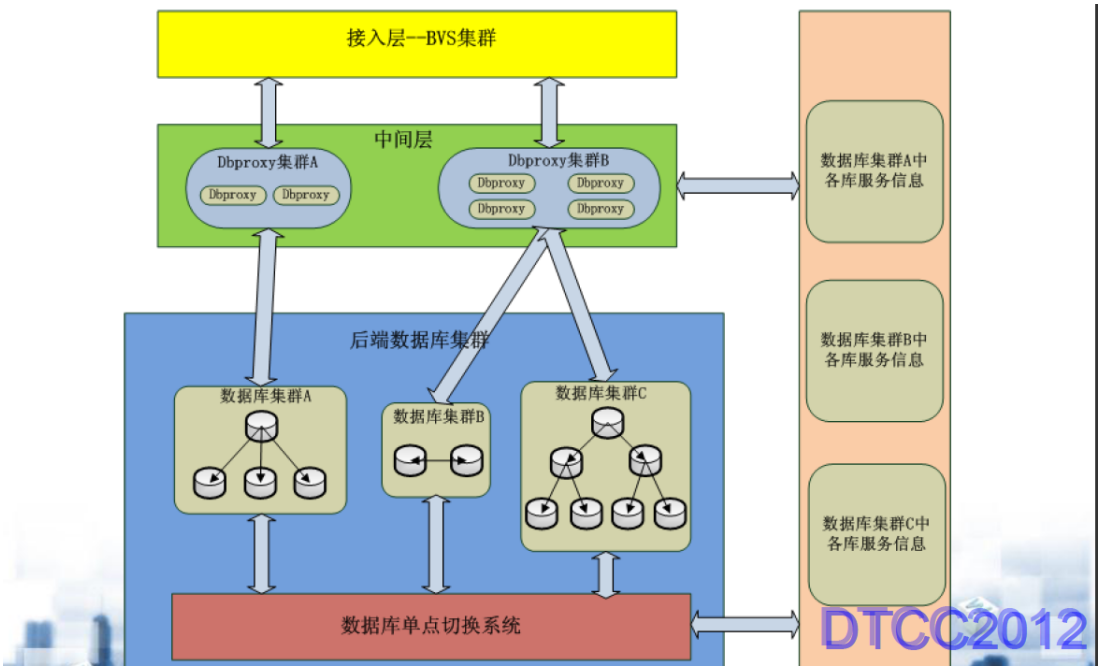
服务器+复制级别的 failover。自动识别复制角色，监控和故障转移。新 master 的选举和集群配置修改。

需要单独 server 作为 manager 节点。配置中需配置各 server root 权限，建立集群内公钥认证。新 master 的选举的判断依据为复制进度，未考虑 server 的负载能力。

f) Keepalived 等。

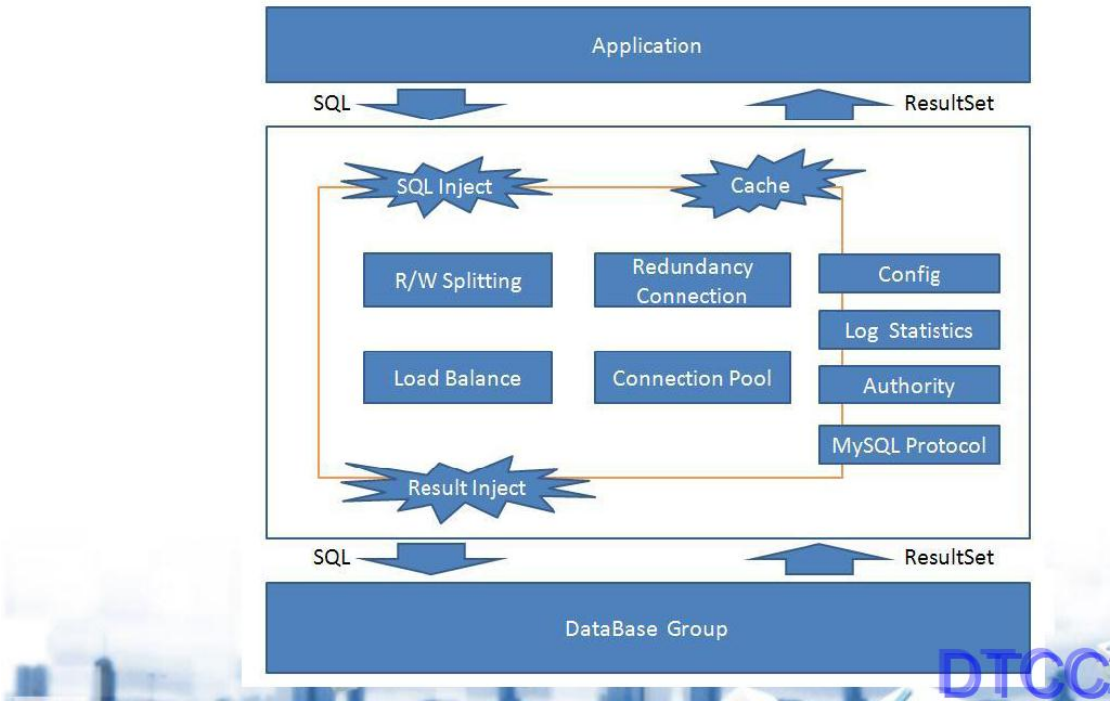
百度中间层技术(dbproxy)

整体架构



定位：对应用透明、高并发、低延迟

逻辑框架



功能模块

1.读写分离

a．读写请求的划分规则。

关键词划分：select/update/delete/insert。（select … for update）

枚举读操作 sql。

b．会话一致性，一个会话写请求结束后在一定时间内读请求需要访问主库。（复制的延时）。

c．事务支持。通过事务关键词获取事务。（start transaction/begin/set autocommit=0）

d． 会话状态保持。会话中 sql 别分离后（use db/set names）等状态保持。

2.负载均衡

a．负载算法为基于数据库当前连接数

b．新建连接—选取集群中同类角色中“当前连接数/权重”最小的数据库。

3.连接池

A．三级 hash 策略（read/write,用户权限，session 属性），保证连接可用

4.部署维护

a.配置信息热加载。

b.支持多个数据库集群

c.自身压力

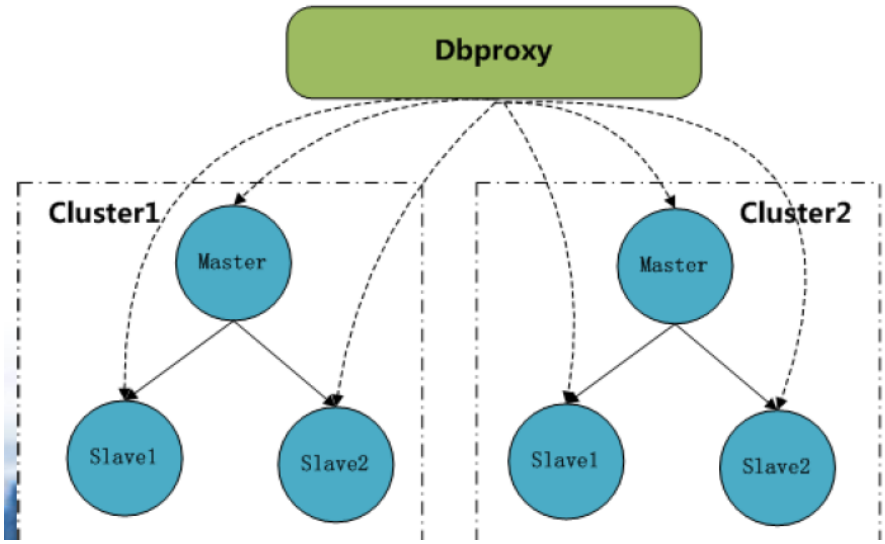
d.流量控制

5.性能

单进程 QPS <17000

Query 通过 dbproxy 延迟：百微秒级

Cpu 消耗<1%,内存消耗 200M



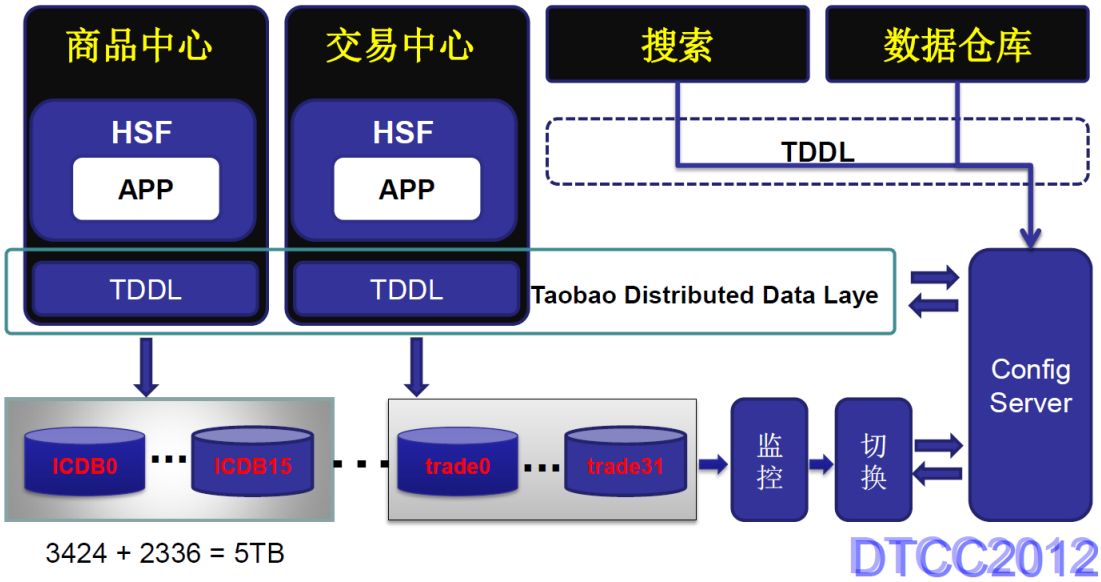
淘宝 tddl

Qps：用户库，message，商品库，物流库，评价库。

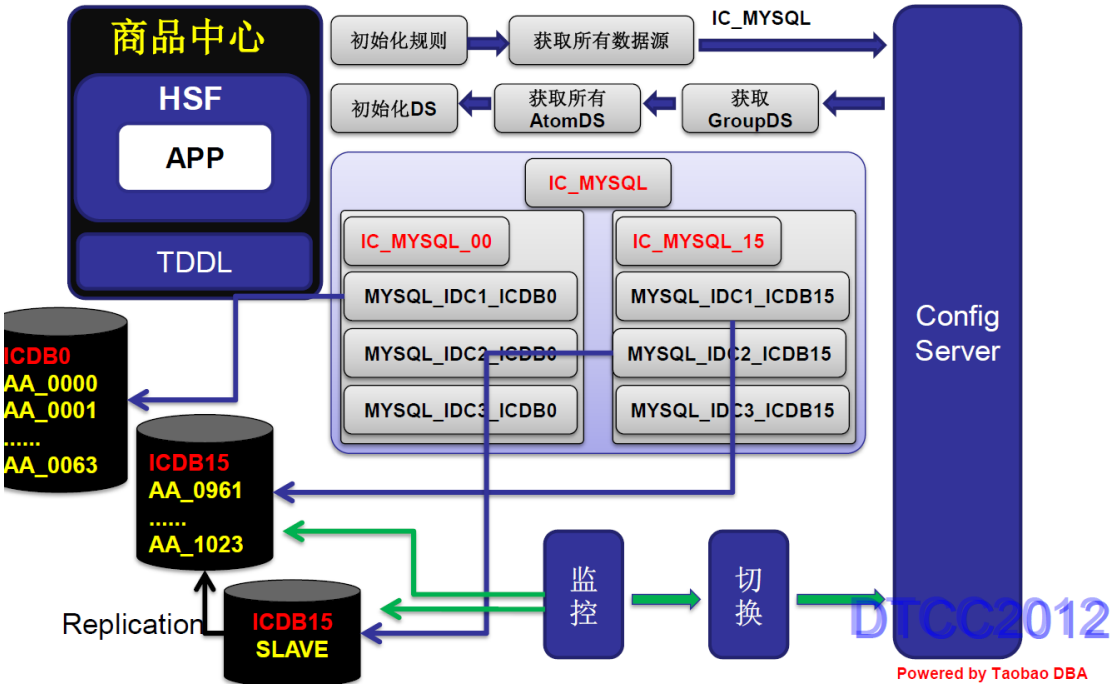
197010+9519+284006+7700+13195+3161

= 51 4591 (2011-11-11/12-12)

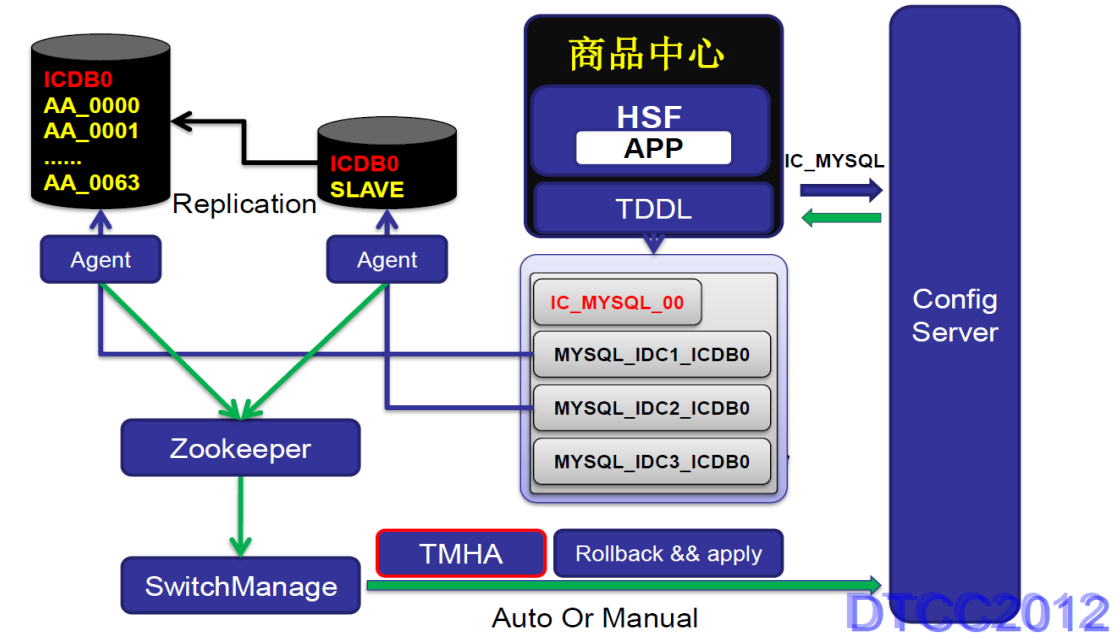
整体架构



逻辑结构



switch



TMHA：对 MHA 中 master 的 rollback 优化。根据 slaves 的复制进度，分配不同重做日志，选举最佳 slave 成为 master。