构建高性能与高可用性的MySQL中间层之路

朱 超

2013年3月17日

内容提纲

- 背景
- 架构与原理
- 改进点
- 新功能
- 未来发展

应用程序员需要关注很多DB的细节

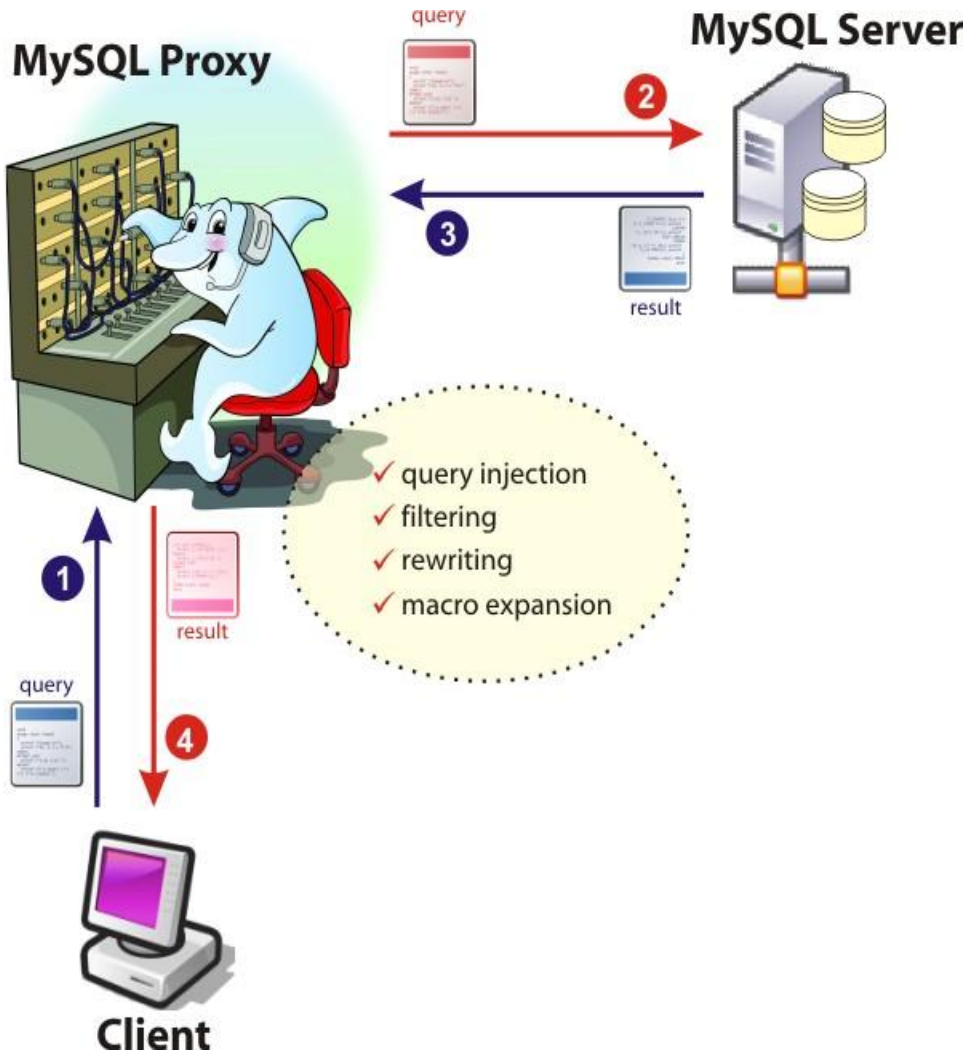
- 配置主库与多个从库的IP和端口
- 自己实现读写分离
- 自己实现分表

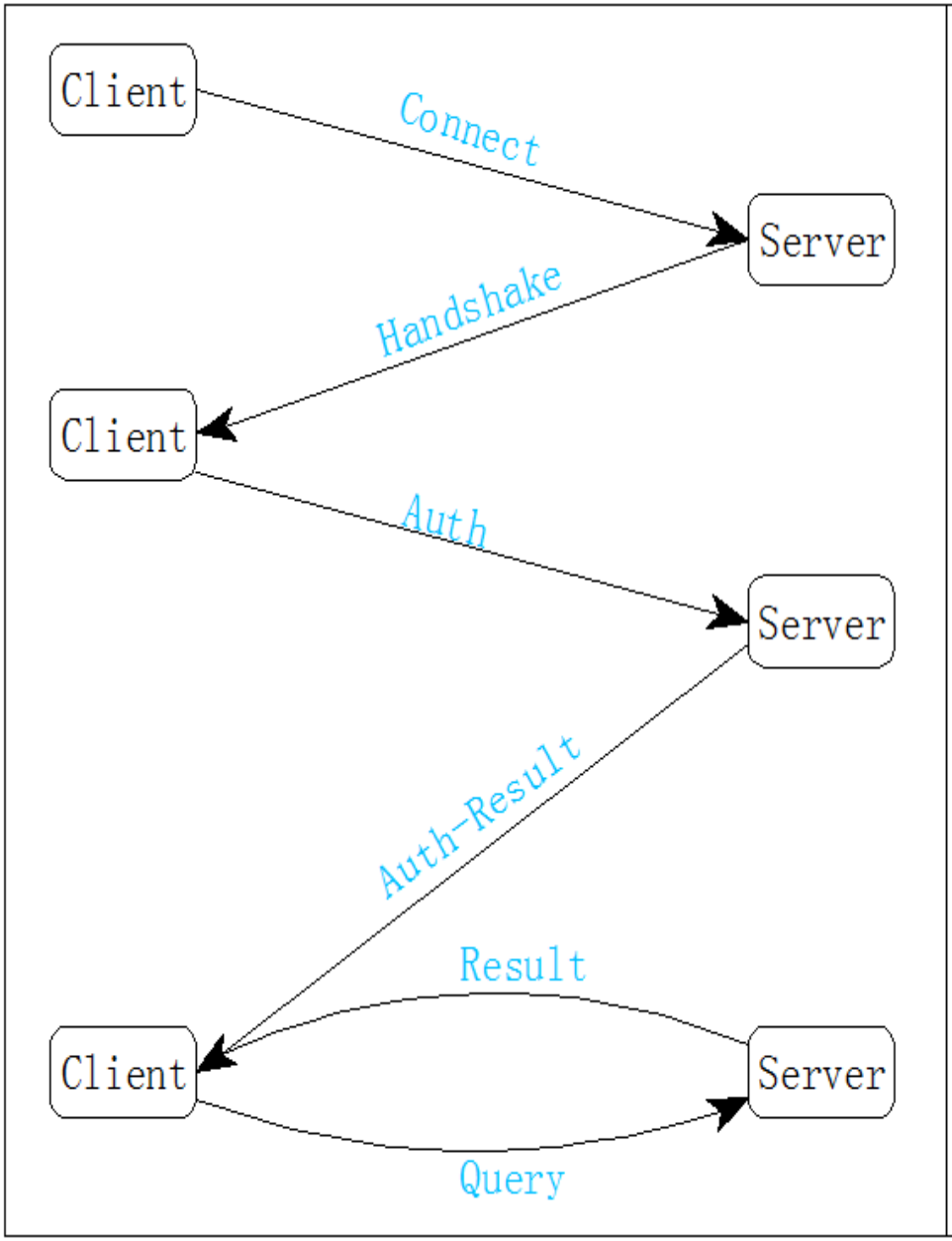
DBA的运维工作繁重

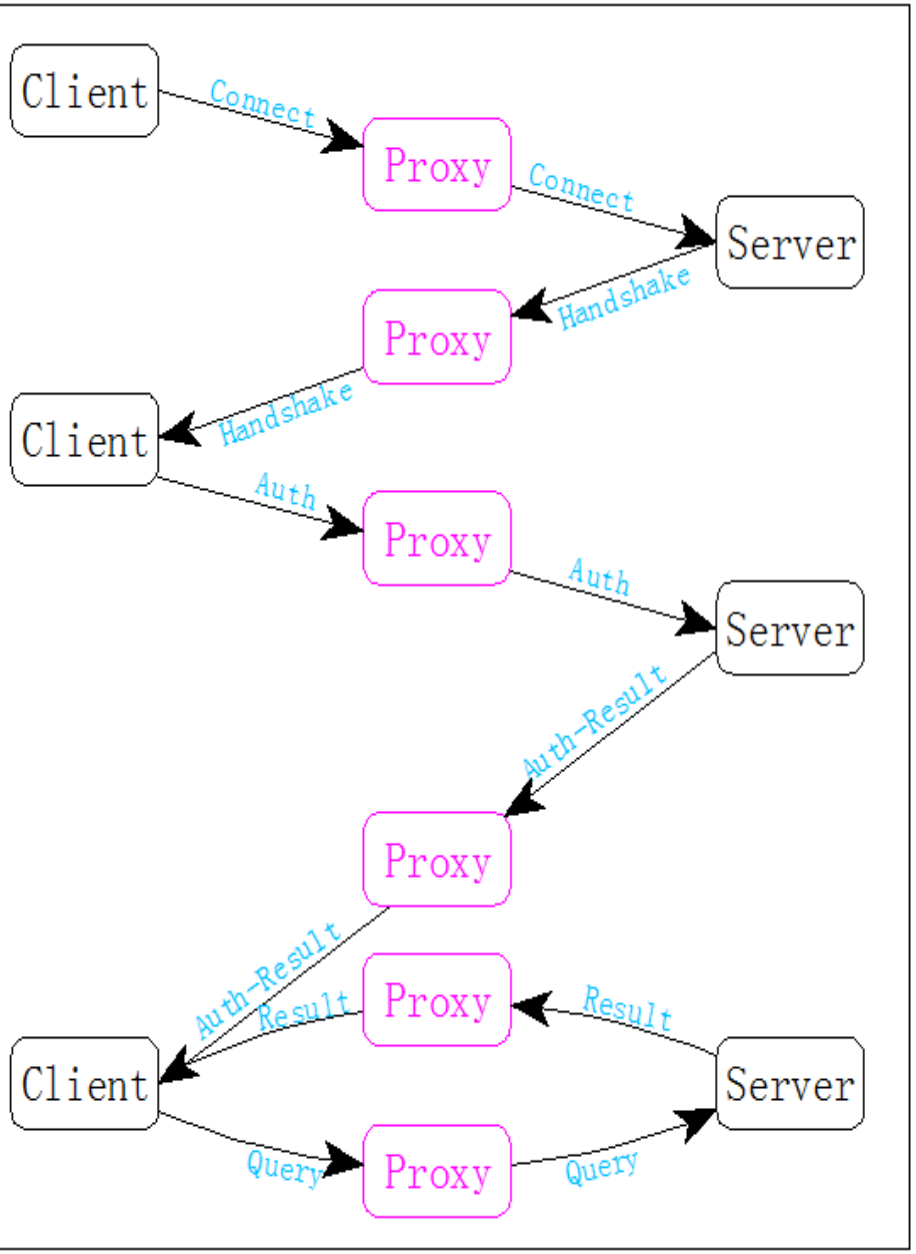
- DB上线与下线
- 协调应用修改配置

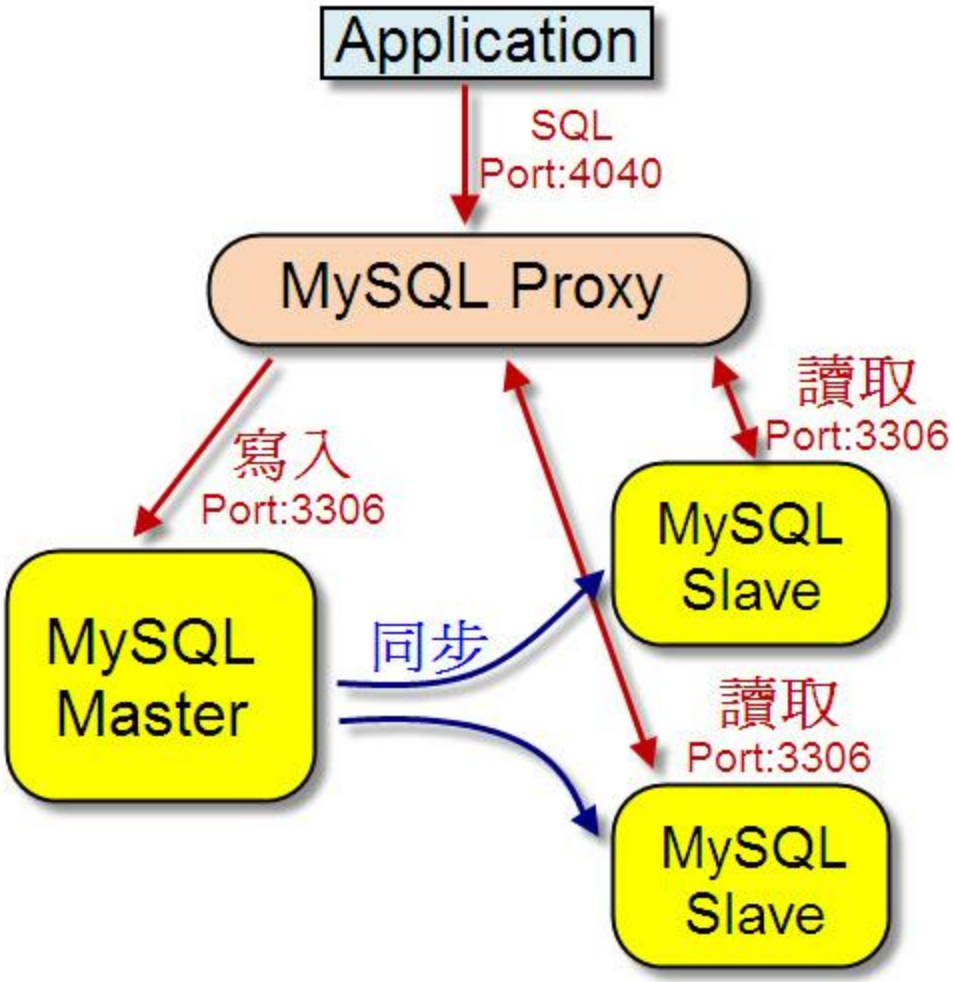
中间层的目标

- 使应用程序员专注于编写业务逻辑
- 减轻DBA的工作量

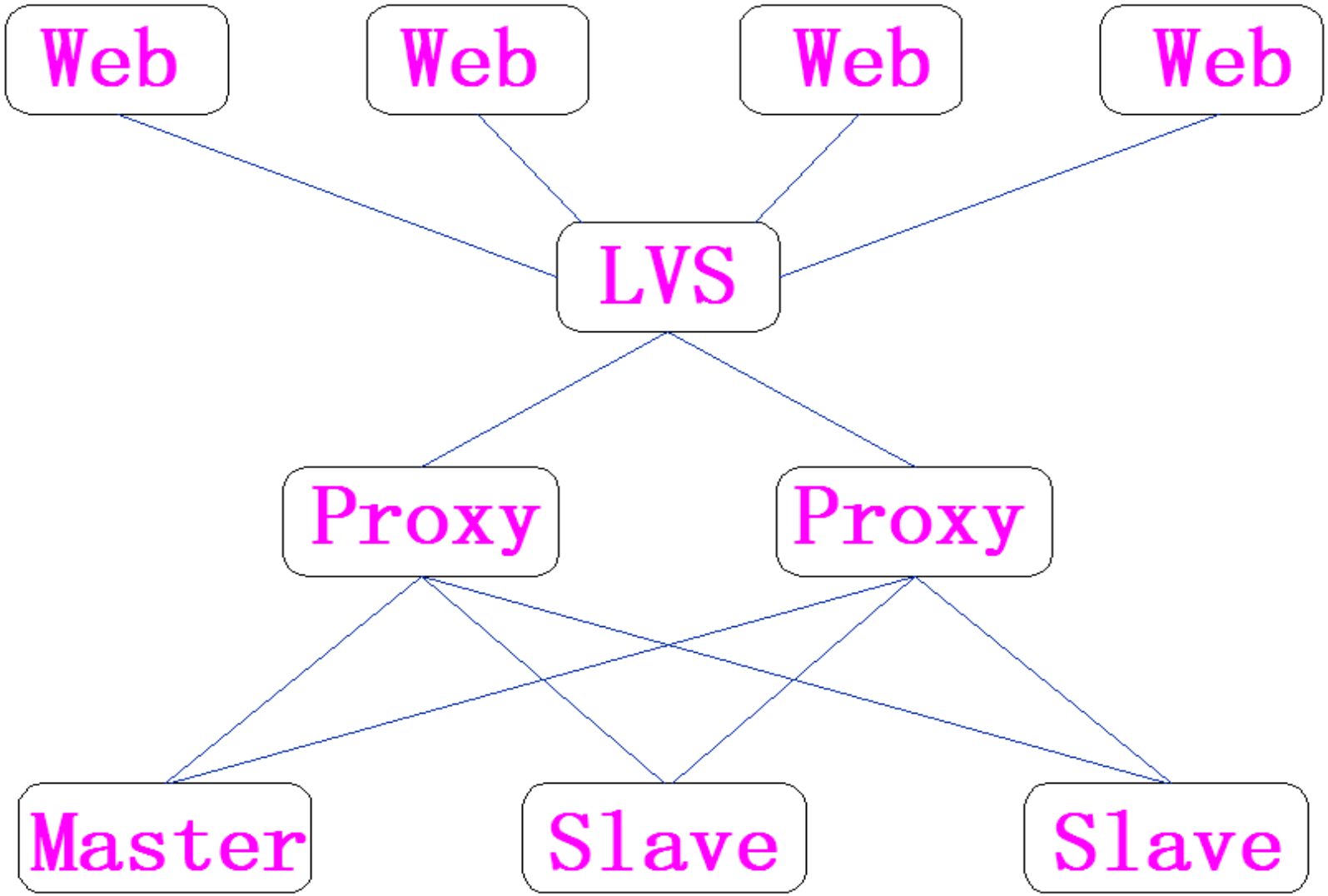

MySQL Proxy
query
MySQL Server
2
3
result
✓ query injection
✓ filtering
✓ rewriting
✓ macro expansion
1
result
query
4
Client


Client
Connect
Server
Handshake
Client
Auth
Server
Auth-Result
Result
Client
Server
Query


Client
Connect
Proxy
Connect
Server
Proxy
Handshake
Handshake
Client
Auth
Proxy
Auth
Server
Auth-Result
Proxy
Auth-Result
Result
Proxy
Result
Client
Server
Query
Proxy
Query


Application
SQL
Port:4040
MySQL Proxy
寫入
Port:3306
讀取
Port:3306
MySQL
Master
同步
MySQL
Slave
讀取
Port:3306
MySQL
Slave

我们目前的架构



改进点

- 主库宕机不影响读
- 连接池
- 多线程
- Lua VS C
- 字符集修正
- 加解锁语句
- 存活检测
- 消除死等
- 新协议兼容
- 线程模型

改进点之1：主库宕机不影响读

- 官方：主库宕机从库亦不可用
- 阶段1：主库宕机时可读不可写

改进点之2：连接池

- 官方：连接池形同虚设，连接数不断上涨
- 阶段1：实现了连接复用
- 阶段2：各线程连接池独立(QPS提高2倍)
- 阶段3：各用户连接池统一

改进点之3：多线程

- 官方：多线程下频繁崩溃
- 阶段1：设置独立的回收线程
- 阶段2：各线程连接池独立

改进点之4：Lua VS C

- 官方：主要的功能逻辑使用Lua脚本编写，效率低
- 阶段1：C改写，QPS提高3倍，latency降低80%
- 阶段2：线程锁粒度细化，QPS提高50%

改进点之5：字符集修正

- 官方：多个客户端分别set不同的字符集，会导致字符集混乱
- 阶段1：自动将服务端的字符集修正为客户端的字符集

改进点之6：加解锁语句

- 官方：不支持get_lock和release_lock，锁权限混乱
- 阶段1：加锁过程中保持当前连接

改进点之7：存活检测

- 官方：利用正常请求的执行结果判断DB状态，对应用有影响
- 阶段1：利用独立的检测线程判断DB状态

改进点之8：消除死等

- 官方：某台DB无法连接时会僵死
- 阶段1：重新添加事件时指定超时时间

改进点之9：新协议兼容

- 官方：5.5.7以上版本MySQL出现Unknown Command错误
- 阶段1：连接时伪装成5.5.7以下版本的客户端

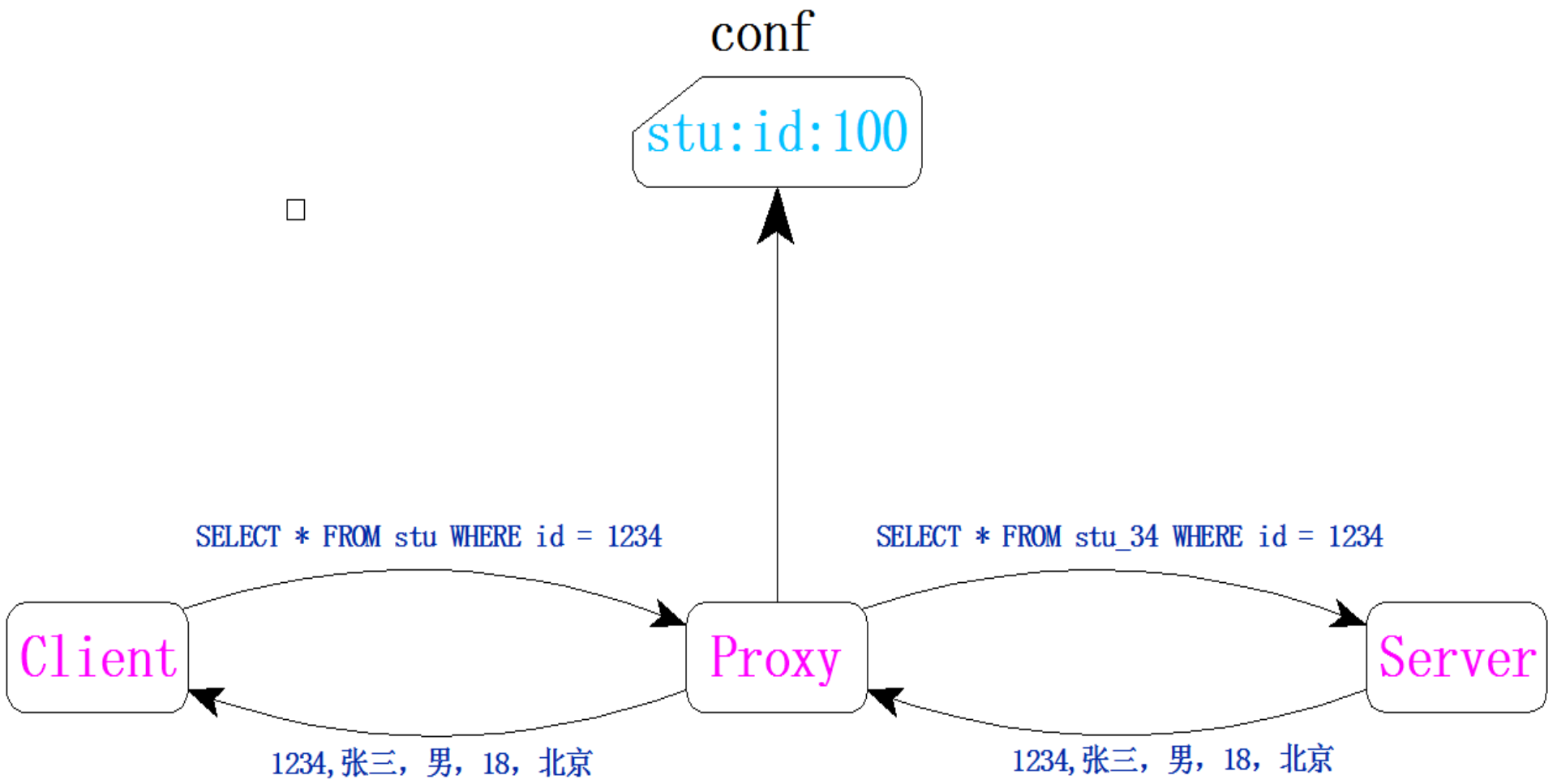
改进点之10：线程模型

- 官方：所有线程监听同一个fd，惊群问题
- 阶段1：每个线程监听自己的fd
- 阶段2：同一个SQL请求由单个线程完成
- 阶段3：所有线程均可接受连接并处理

- 分表
- 强制读主库
- 负载均衡
- 在线增减与上下线DB
- 平滑重启
- SQL过滤
- IP过滤
- 查询日志

新功能之1：分表

- 需带有分表字段
- 支持SELECT、INSERT、UPDATE、DELETE、REPLACE语句
- 支持多个子表查询结果的合并和排序


conf
stu:id:100
SELECT * FROM stu WHERE id = 1234
SELECT * FROM stu_34 WHERE id = 1234
Client
Proxy
Server
1234,张三，男，18，北京
1234,张三，男，18，北京

新功能之2：强制读主库

- /*master*/ SELECT * FROM mytable

新功能之3：负载均衡

- 精确到每个SQL请求
- 每台从库被赋予一个权重
- 未来支持动态调整权重

新功能之4：在线增减DB

- select * from backends(官方)
- add mastr ip:port
- add slave ip:port@weight
- remove backend i
- set offline i
- set online i

新功能之5：平滑重启

- 阶段1：修改配置文件中的online标志
- 阶段2：发信号

新功能之6：SQL过滤

- 黑名单
- 白名单

新功能之7：IP过滤

- 精确IP
- IP段

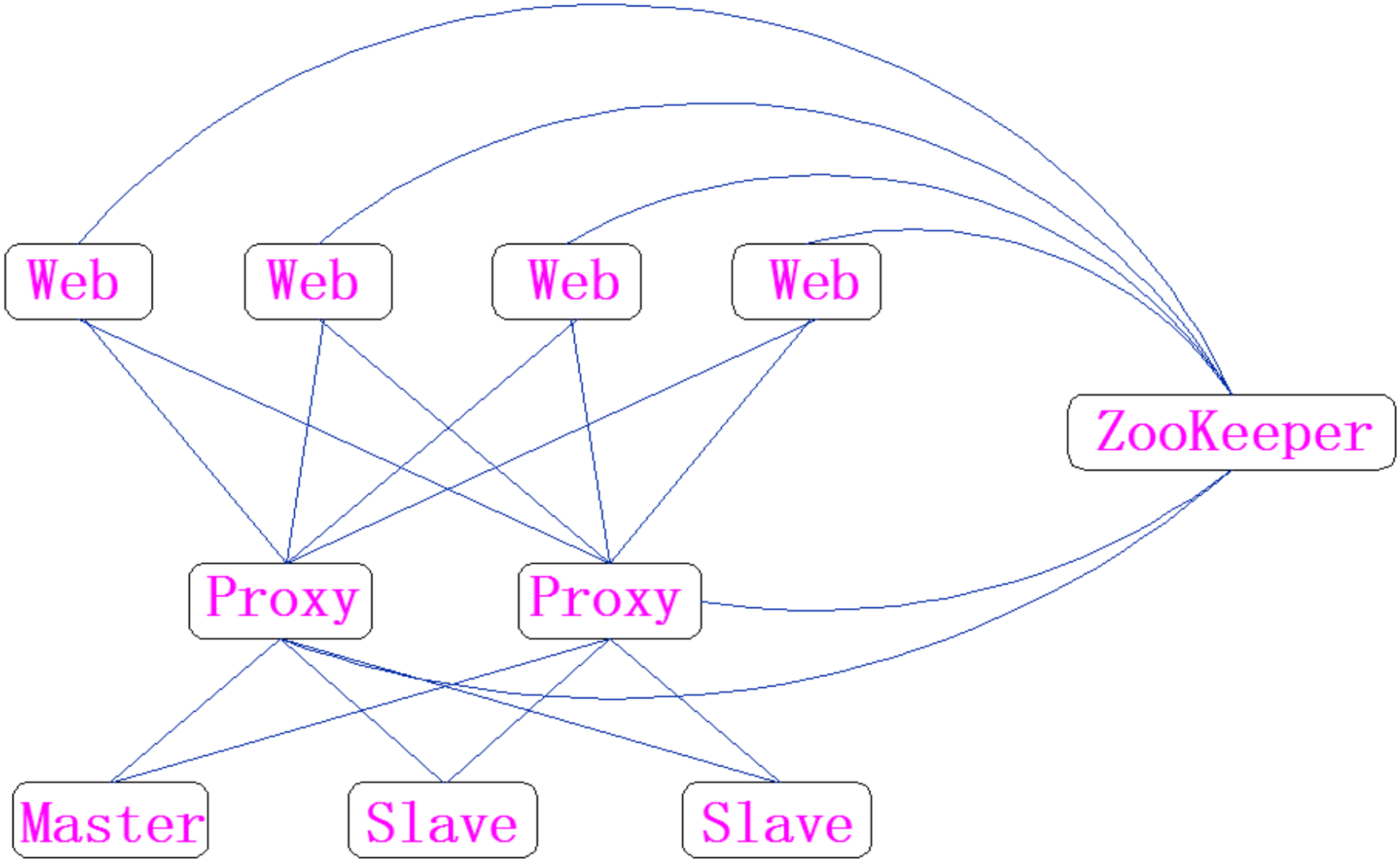
- 记录所有处理的SQL语句，包括客户端IP、实际执行该语句的DB、执行成功与否、执行所耗费的时间

- [02/14/2013 16:21:41] C:192.168.1.2 S:192.168.1.3 OK 0.807 "SELECT * FROM person.mt WHERE id = 1025189561"

未来发展

- 引入zookeeper
- 完全的自验证
- 数据分片

引入ZooKeeper



Thanks
Q & A