淘宝mysql数据库高可用的设计实现-TMHA

穆公(朱金清)

mugong.zjq@taobao.com
微博： 淘穆公
阿里DBA团队博客：http://www.taobaodba.com/

大纲

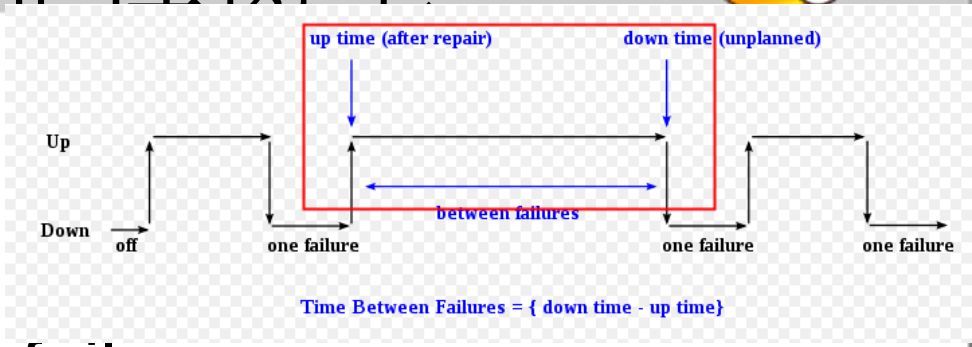
- MySQL高可用的难题
- TMHA的整体设计
- TMHA如何实现异常切换
- TMHA如何保证数据一致性
- TMHA如何实现自动切换
- TMHA如何解决主备库延迟
- 总结

背景

- 互联网应用以普通的PC服务器为主
- 免费的开源软件: Linux平台、MySQL

- MySQL数据库的主要问题
  - 主库单点问题
    - 通过业务功能的写入主库通常只能有1个
    - 除非应用程序自己完成容灾

背景-可靠性衡量



- 可靠性指标MTBF
  - Mean Time between failures
- 1million hours的含义
  - 10,000台服务器同时运行100小时就会坏一台
- 服务器主要部件MTBF
  - 主板、CPU、硬盘 1million hours (厂家标称值)
  - 内存 4million hours(8根内存 ~ 1million hours)
- 整体的MTBF~1million/4=250000h~1万天
  - 年故障率约2%-4%

故障率较高

Ref URL: http://en.wikipedia.org/wiki/Mean_time_between_failures
http://wenku.baidu.com/view/7943585c3b3567ec102d8a0f.html

MySQL常用容灾方案—复制

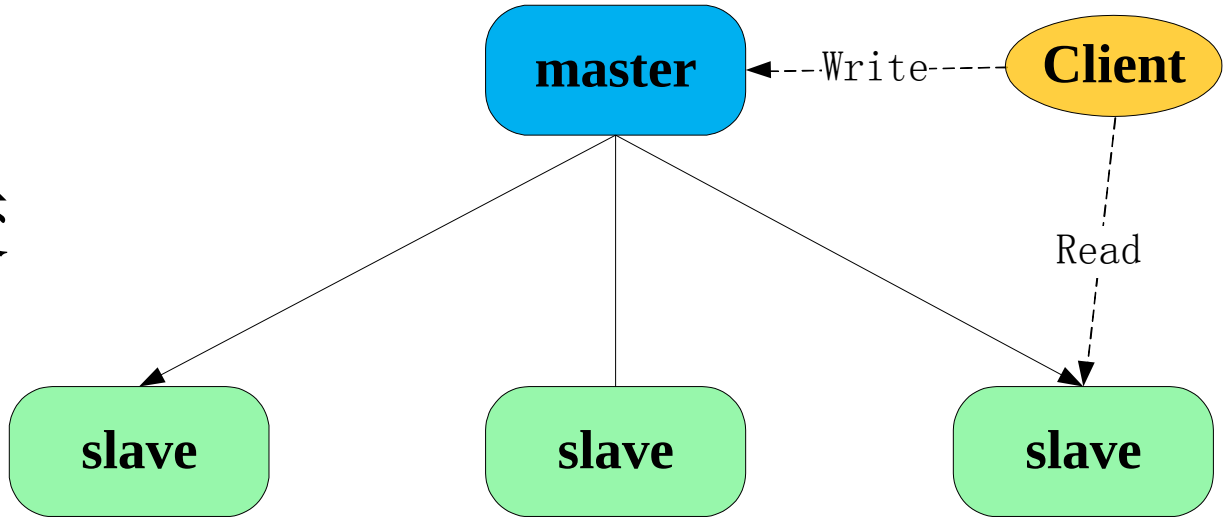
Master :

- 数据发生改变
- 记录变化

Slave:

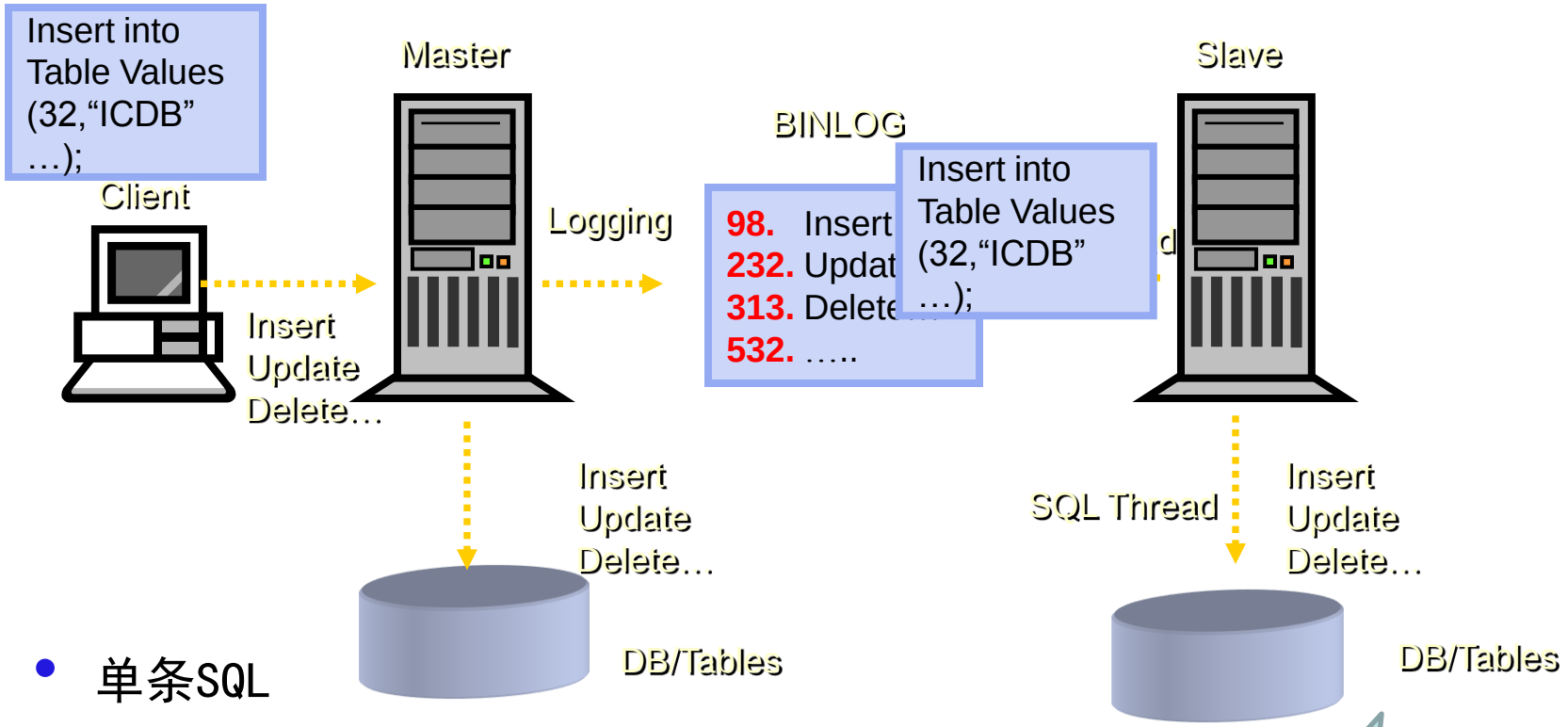
- 获取master的改变
- 同步这些变化

Binary-log



数据有延迟

MySQL复制的演示、延迟

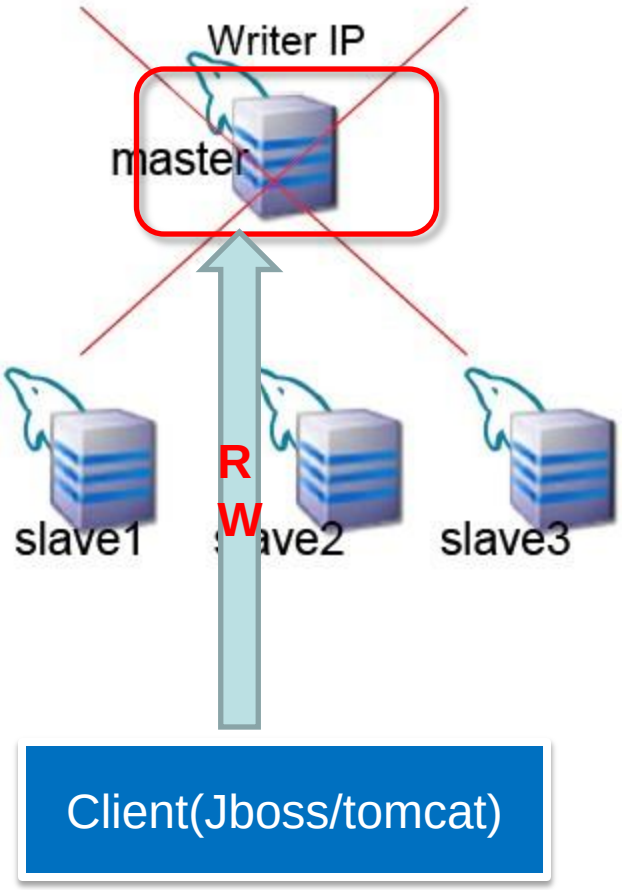


- 单条SQL
  - 执行(执行时间为*T*)完直接写入binlog
  - 延迟大概为*T*
- 一个事务（包括*N*条）
  - 先缓存到cache，全部执行完写入
  - 延迟为*NT*

数据一致性

背景—mysql切换

- master挂了，如何？
  - 选择新的主库
  - 通知应用切换
  - master恢复之后，如何同步
- **着重问题：**
  - 故障是存在的
  - MS数据的一致性保证
  - 新主库的选举 / 应用程序感知



DB/WEB
同时切换

大纲

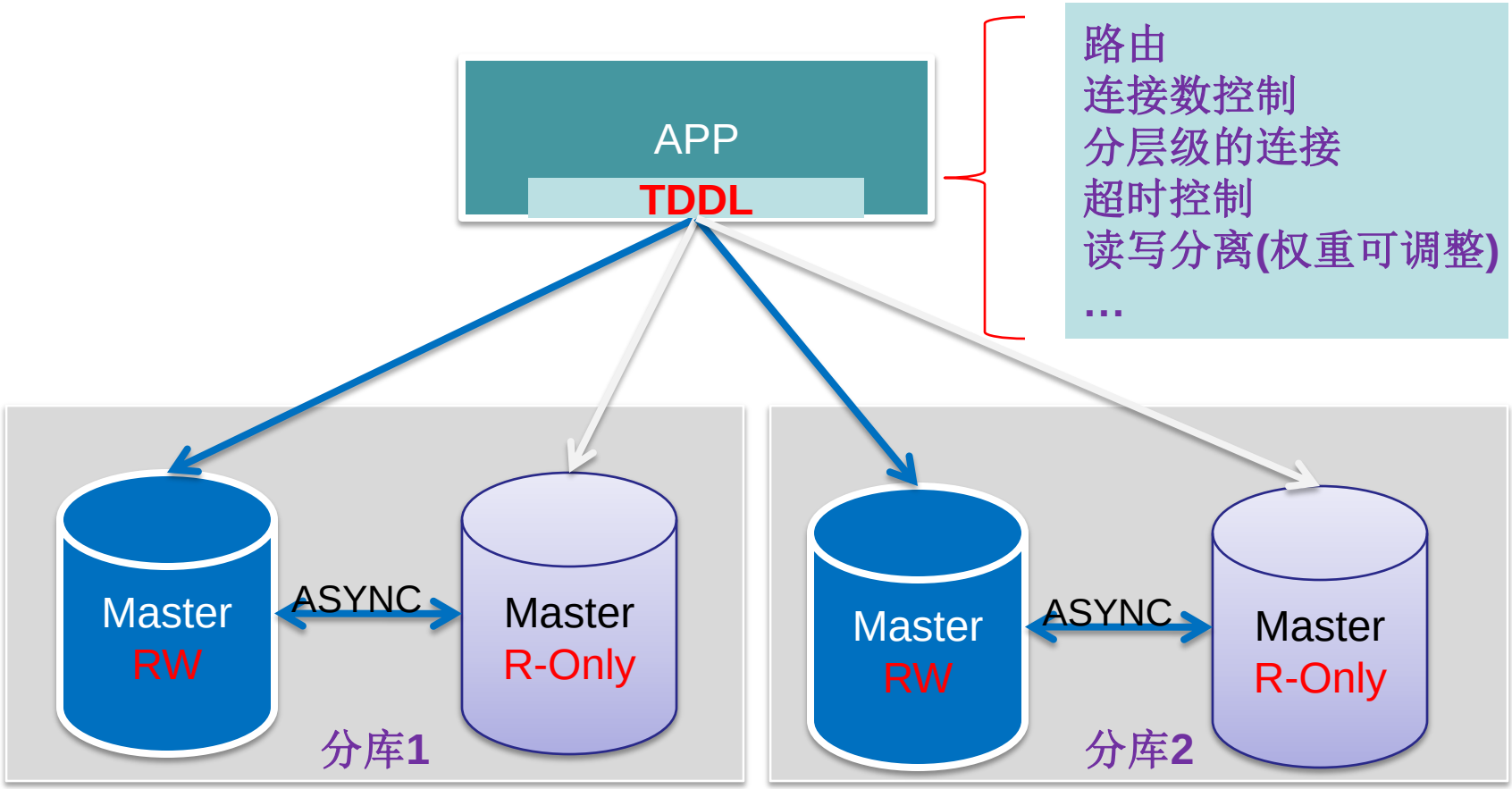
- MySQL高可用的难题
- TMHA的整体设计
- TMHA如何实现异常切换
- TMHA如何保证数据一致性
- TMHA如何实现自动切换
- TMHA如何解决主备库延迟
- 总结

对应用透明的常用方法

- Master采用**虚IP**的方式
  - **前提：**备库与主库在同一网段
  - 阿里云的RDS、云聊PHPWind [1]
  - 腾讯的CDB[2]
- DB对外的接口是**DNS**
  - 优势：备库与主库可以在不同机房
  - 缺点：受限于DNS，若DNS故障，服务不可用
- MHA[3]: 多个从库之间选择一个主库

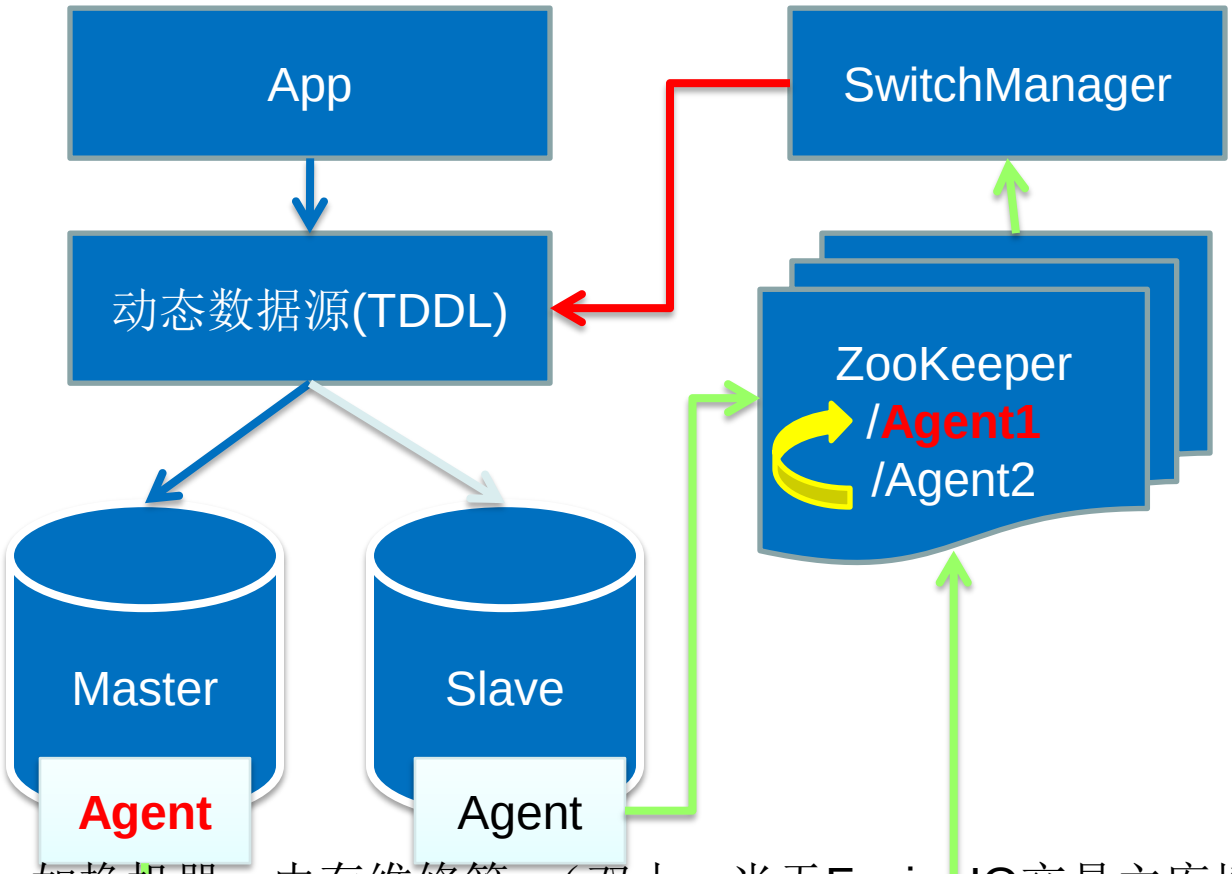
- [1] http://app.phpwind.com/
- [2] http://wiki.opensns.qq.com/wiki/CDB
- [3] http://code.google.com/p/mysql-master-ha

分布式数据中间层(TDDL)



1. Master和Master-Readonly的mysql部署在不同机房
2. 异步复制，有数据延迟
3. 分库分表

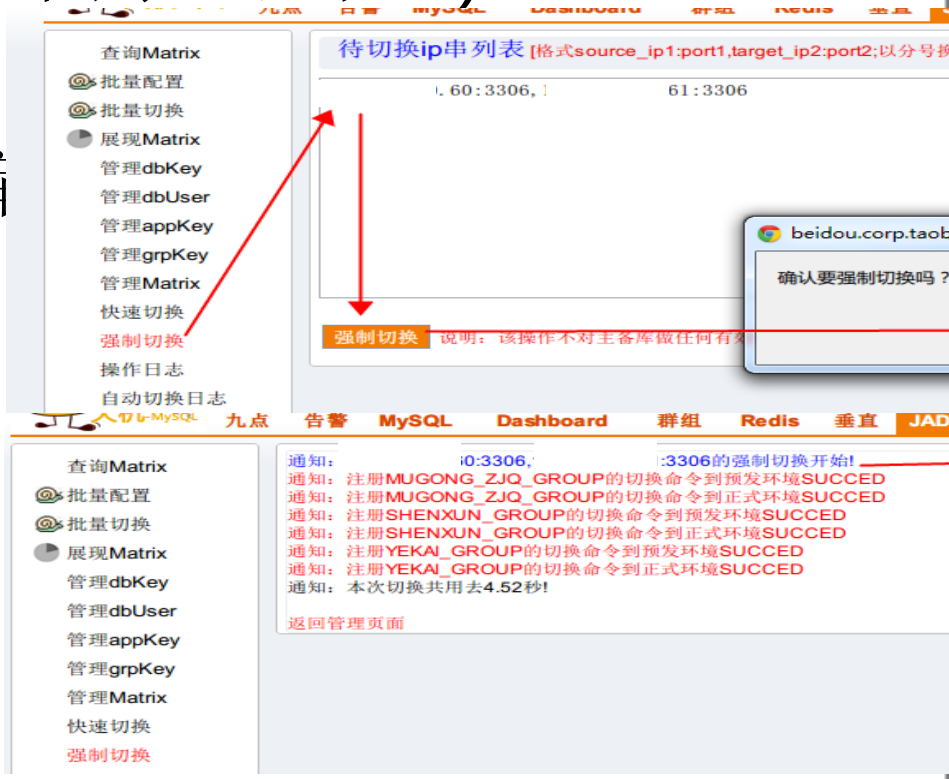
TMHA(master HA)整体架构



A) 维护切换，如换机器、内存维修等 （双十一当天FusionIO交易主库切换）
B) 异常切换：
1) master异常挂掉，zookeeper的agent1结点消失（如果网络，zk感知）
2) agent2得知watcher事件，记录异常，创建异常结点
3) SwitchManager获取最新的异常结点，再次确认是否异常
4) 异常，推送tddl配置，将新主库read-only置为false，即新主库可写

切换类型

- 切换类型
  - 正常切换 (机器维修、扩容等)
  - 强制切换 (主库load非常高，双十一)
  - 自动切换
  - 批量切换 (16、32套库



大纲

- MySQL高可用的难题
- TMHA的整体设计
- TMHA如何实现异常切换
- TMHA如何保证数据一致性
- TMHA如何实现自动切换
- TMHA如何解决主备库延迟
- 总结

分布式系统的异常检测思路

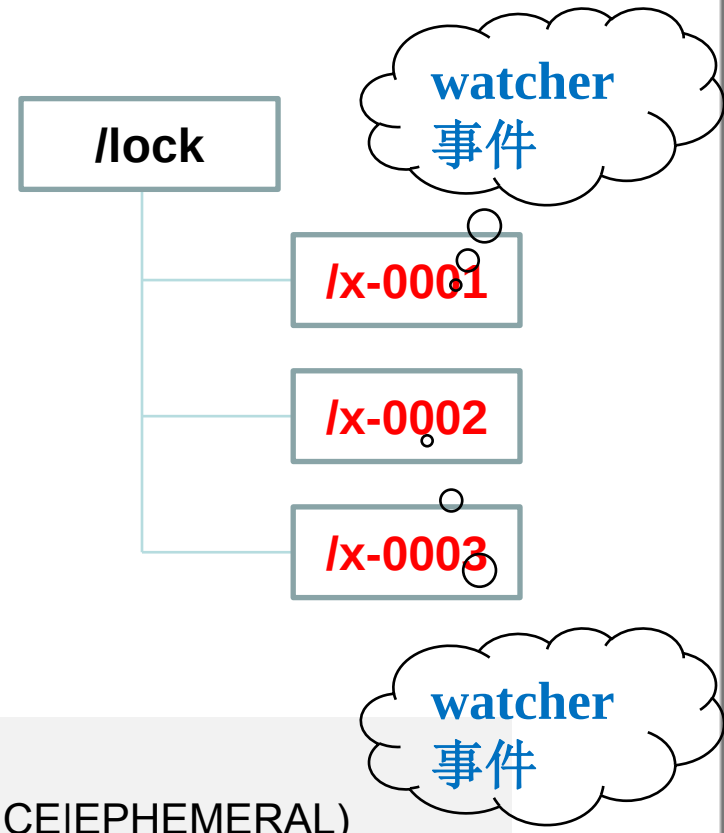
- Paxos：一半机器存活即可
- 实践中，常用master + lease来提高效率

- 分布式系统协调服务
  - Chubby (Google: Bigtable, MapReduce)
  - Zookeeper (Yahoo!: hbase, hadoop子项目)

- [1] The Chubby lock service for loosely-coupled distributed systems （google论文）
- [2] http://nosql-wiki.org/wiki/bin/view/Main/ThePartTimeParliament
- [3] http://hadoop.apache.org/zookeeper
- [4] PaxosLease: *PaxosLease*: Diskless Paxos for Leases

主库选举的检测逻辑

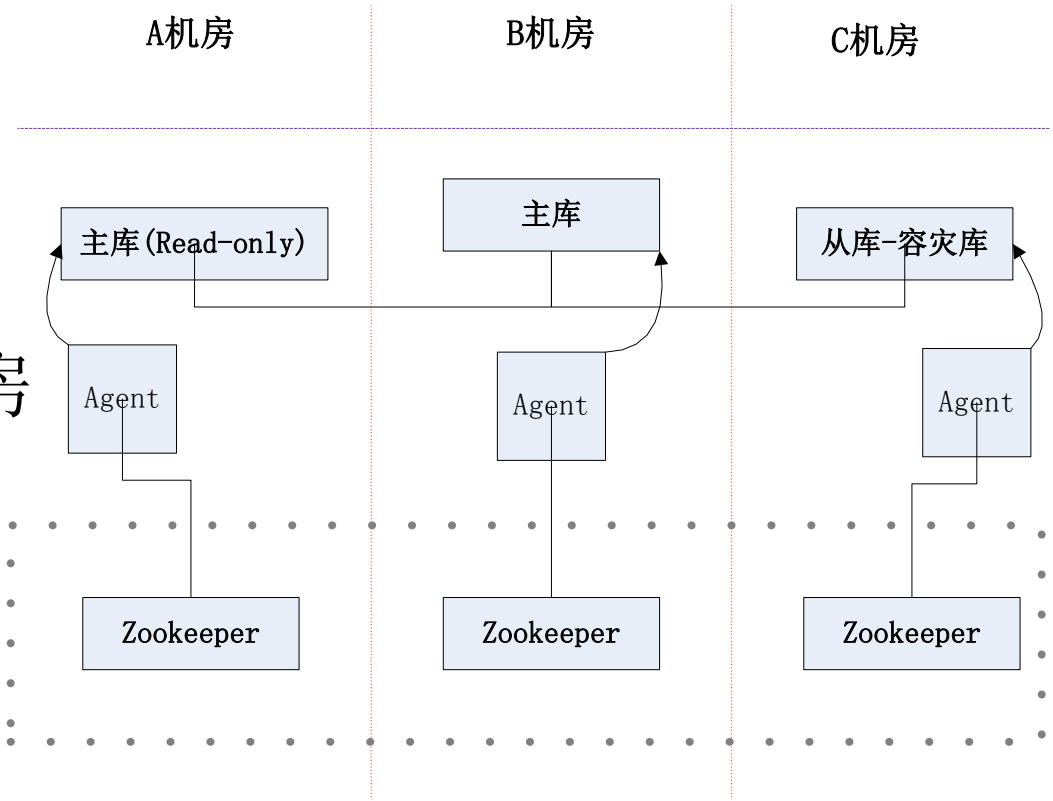
- **主库切换选举 （zk实现写锁）**
  - 每个mysql的客户端对应一个节点
  - 主库对应的节点为第一个节点
  - 若主库挂了，节点消失
  - 发起选举，只有一个节点获得lock 即成为新主库



1. 初始化阶段： 创建/transfer服务节点
2. 创建lock子节点，zoo_create(“ /locks/x-”, SEQUENCE|EPHEMERAL)
3. zoo_get_child(“/lock”, NULL) //不设置watcher
4. 若当前client的id(序列的id)是当前最小的节点，则获得锁，退出
5. 否则，zoo_wexists(last child before id, **watcher**)
   a) 若id不存在，则返回第3步
   b) 等待watcher的触发

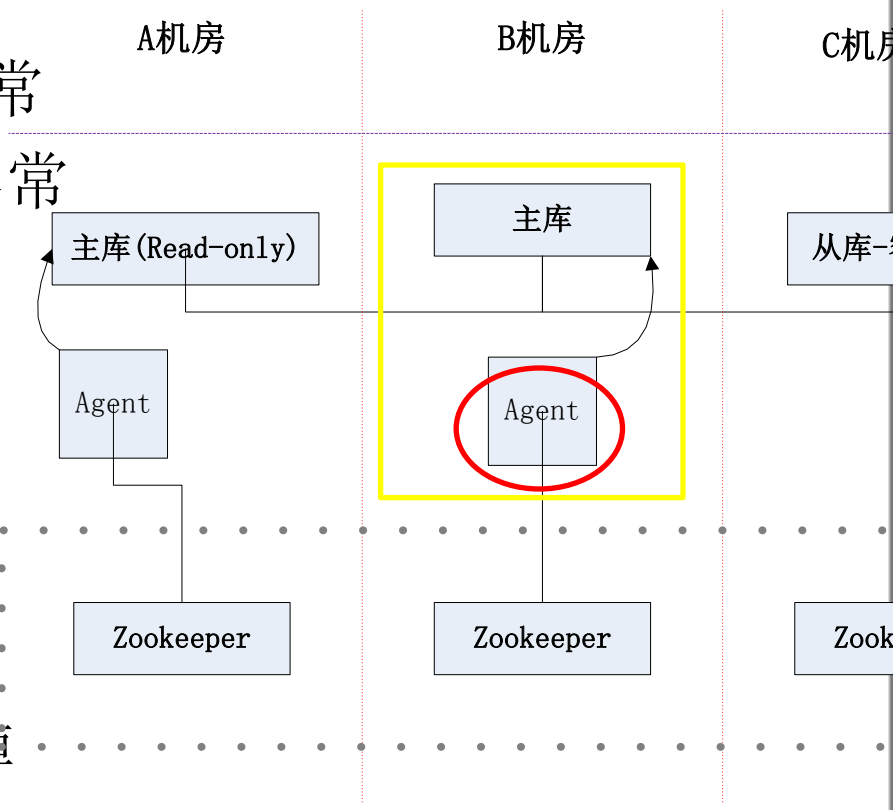
切换部署及使用场景

- 优势
  - 备库与主库不同机房
  - 不受限于DNS
- 场景：三个机房
  - zk部署在三个机房
  - mysql:agent=1:1

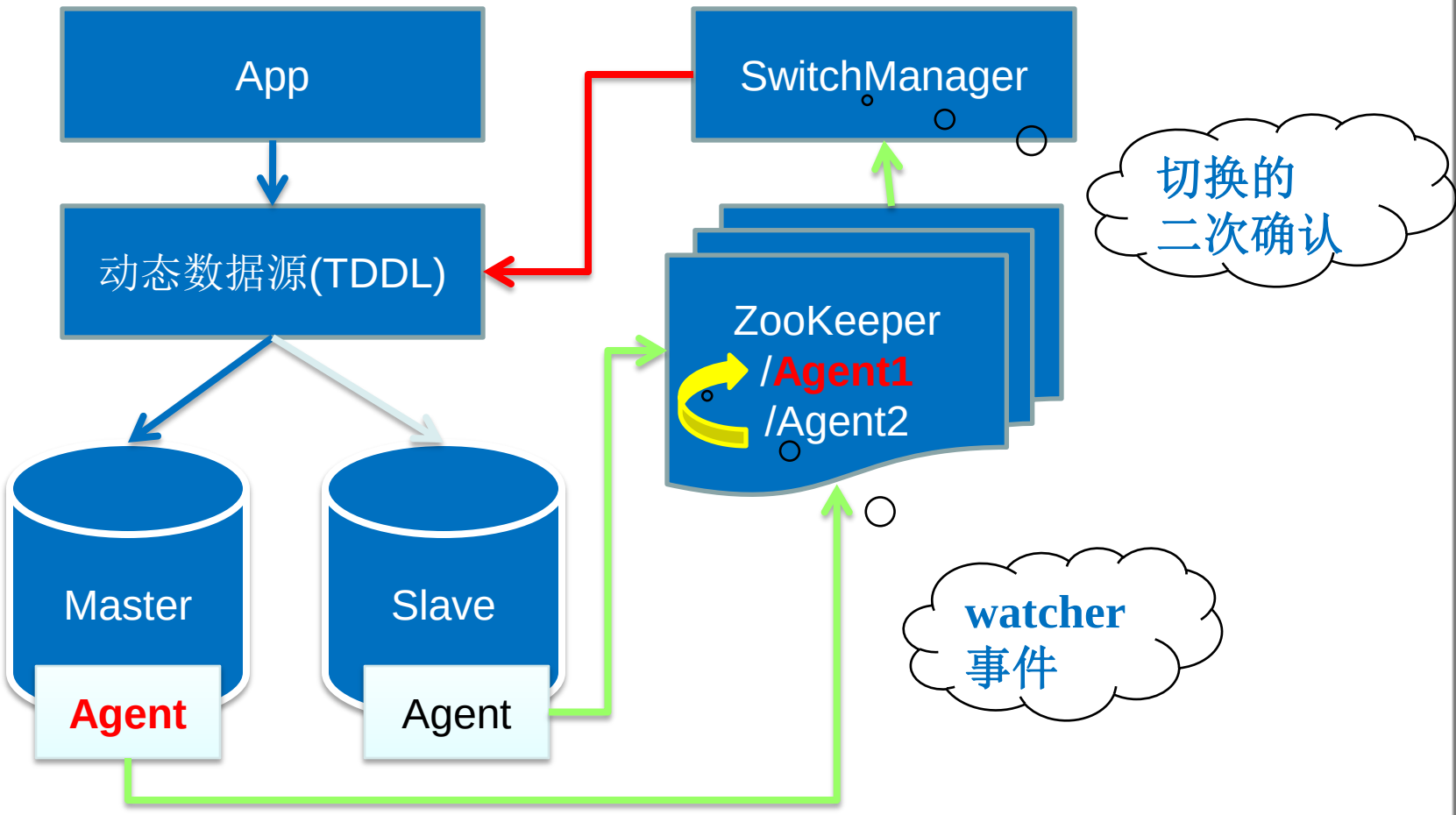


主库切换的触发条件

- agent异常
  - a1:agent异常退出
  - a2:agent与mysql的通信异常
  - a3:agent与zk之间的网络异常
  - a4:机器死机
- mysql数据库
  - m1:访问异常
  - m2:机器死机(同a4)
  - m3:机器的网络异常(同a3)
  - m4:所在的整个机房down掉



触发条件的抽象化、程序化

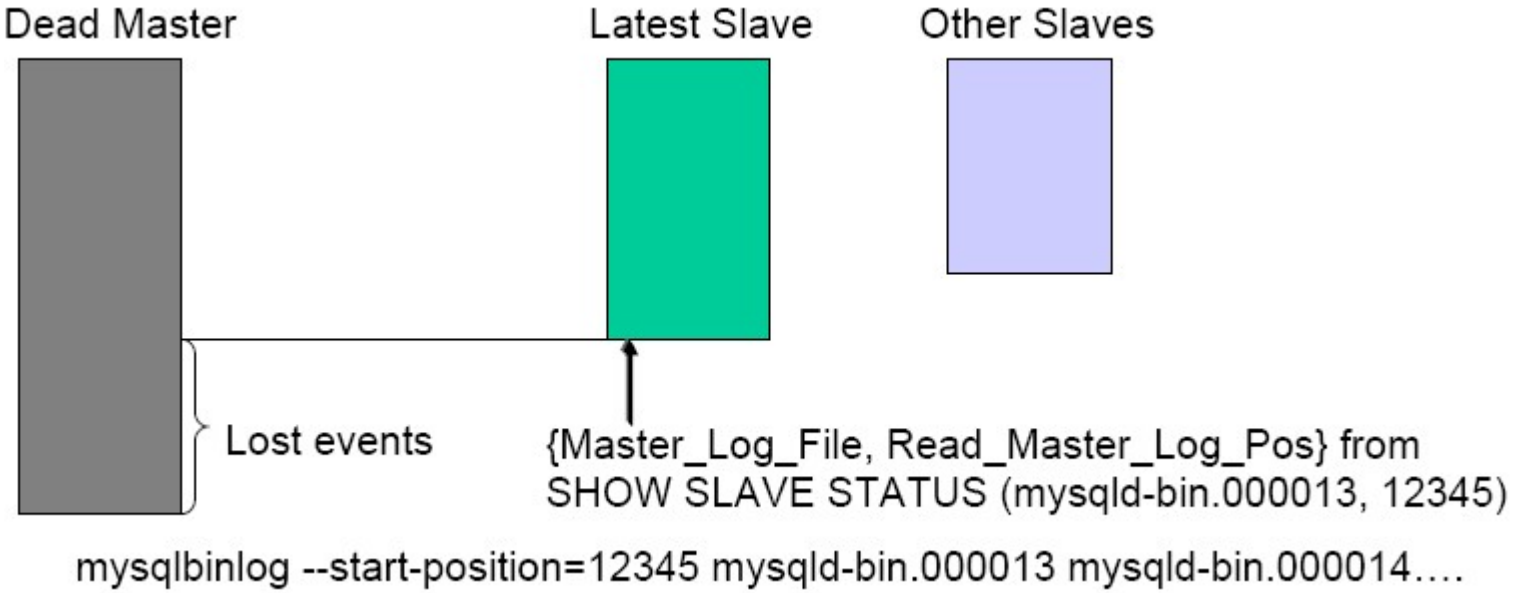


1. 所有条件的表现都是/Agent1结点消失
   - mysql异常，agent1主动删除结点
   - zk/网络异常，达到zk的超时后消失
2. Agent2得到Agent1消失的事件(zookeeper的Watcher机制)

大纲

- MySQL高可用的难题
- TMHA的整体设计
- TMHA如何实现异常切换
- TMHA如何保证数据一致性
- TMHA如何实现自动切换
- TMHA如何解决主备库延迟
- 总结

一致性—可能丢失的数据

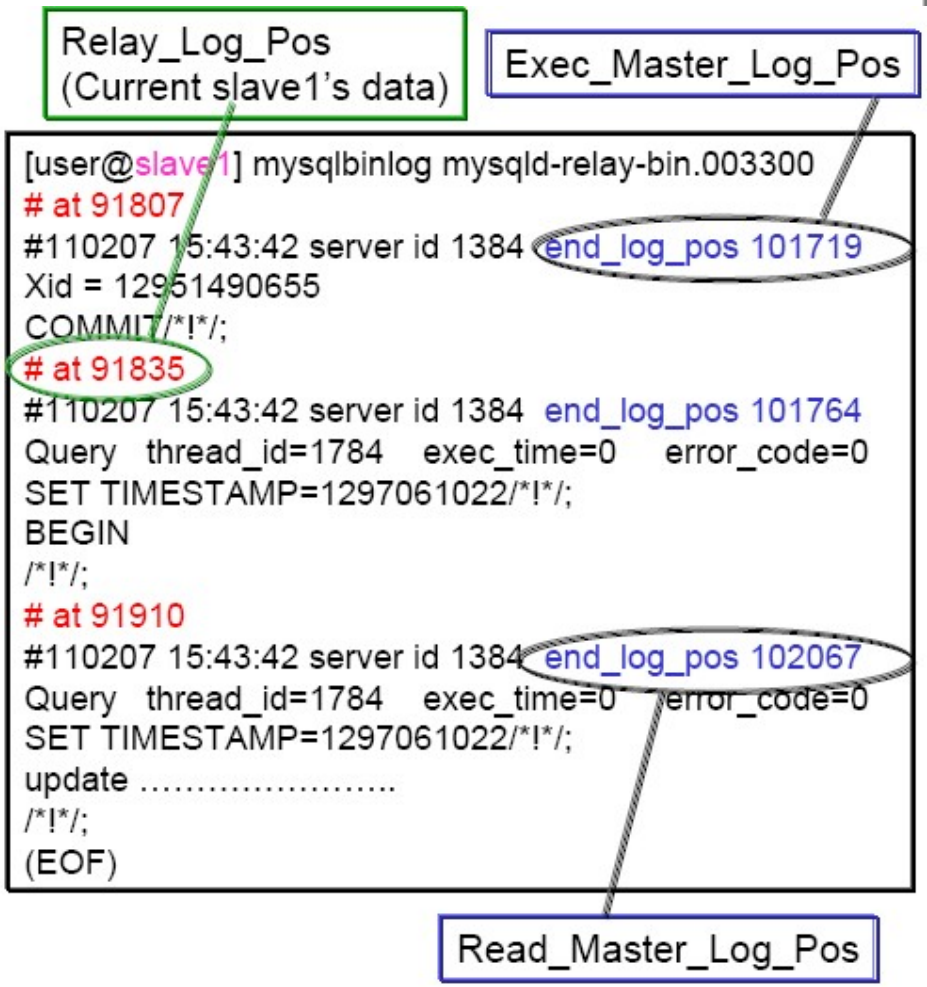


- 挂掉的master的binlog能否获取到 (记做Δ1)
- Slave机器上的relay-log损坏(记做Δ2)
- 简称delta(Δ)

REF : http://code.google.com/p/mysql-master-ha/

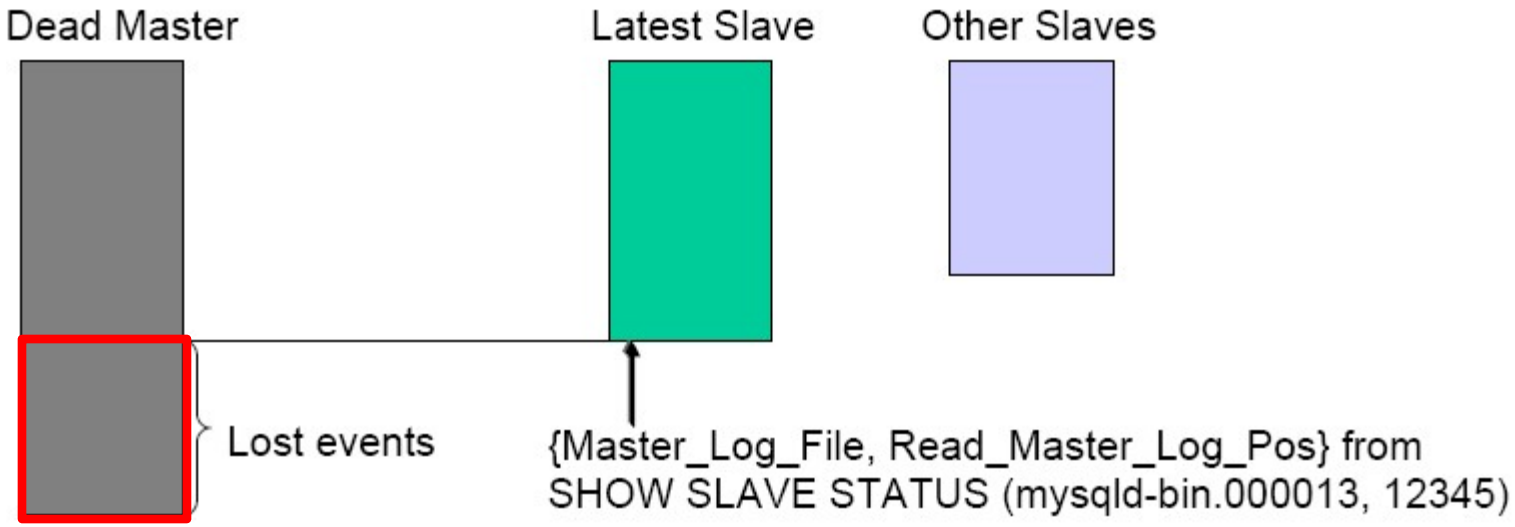
Δ2的回补策略

- Slave的relay-log损坏
  - 判断Exec/Read的pos
  - 若不相等可能有丢失
- 处理方案：
  - reset slave/change ma
  - relay-log重新获取即可



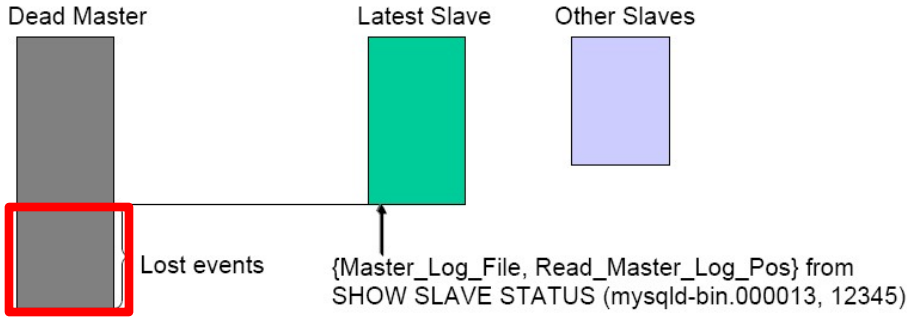
Δ1的处理策略

- 需要根据决策来决定
  - Dead主库起来， **Δ1继续同步**，不切换
  - 切换， Dead主库起来，**主库回滚Δ1**



Δ1的处理策略-回滚

- 回滚宕机主库日志（必须是row模式）



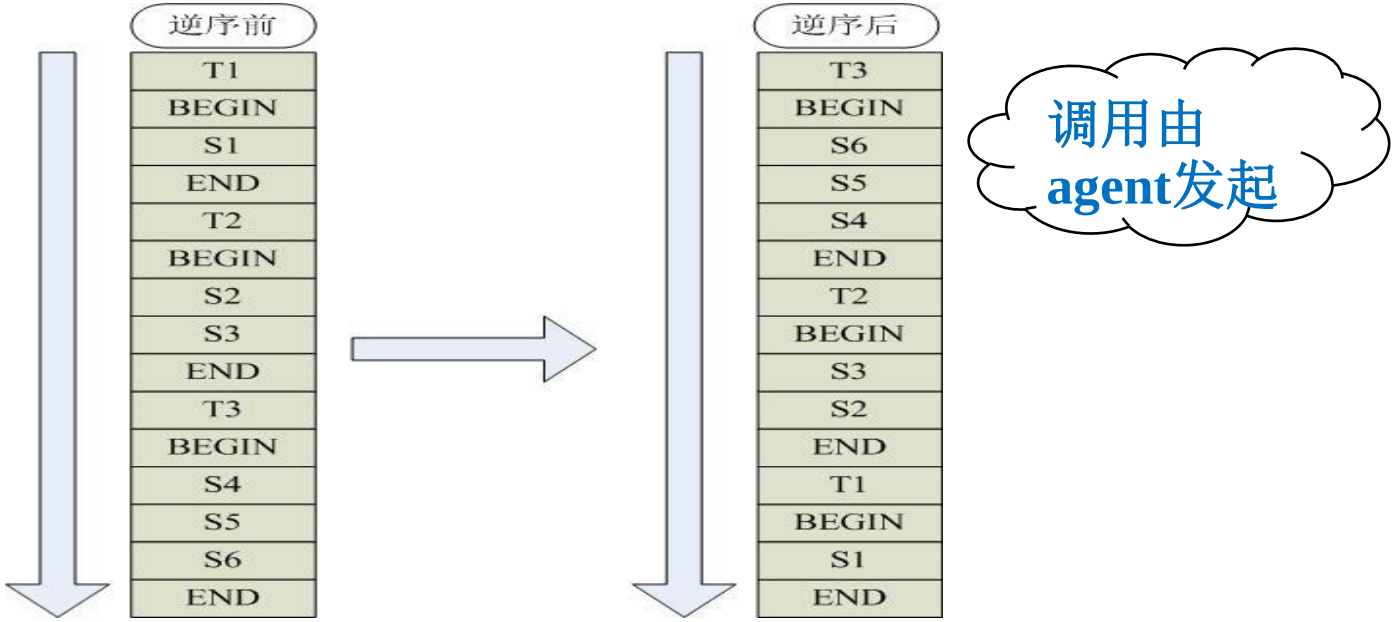

```
/*!*/;
# at 764
# at 805
#120214 17:02:50 server id 54  end_log_pos 659  Table_map: `test`.`ma` mapped to number 33
#120214 17:02:50 server id 54  end_log_pos 703  Write_rows: table id 33 flags: STMT_END_F
### INSERT INTO test.ma
### SET
###   @1=10
### INSERT INTO test.ma
### SET
###   @1=20
### INSERT INTO test.ma
### SET
###   @1=30
```

逆sql：

DELETE FROM test.ma WHERE id=10;

Δ1回滚的原理(rollback.pl)

- 倒置binlog中所有SQL顺序，保证逻辑相反



- 注意：
  - 适当修改mysqlbinlog工具
  - 双字节，第一个字节超过7F,第二个字节为5C
  - URL: http://www.taobaodba.com/html/1520_binlog-to-recover.html

Δ1回滚的使用场景

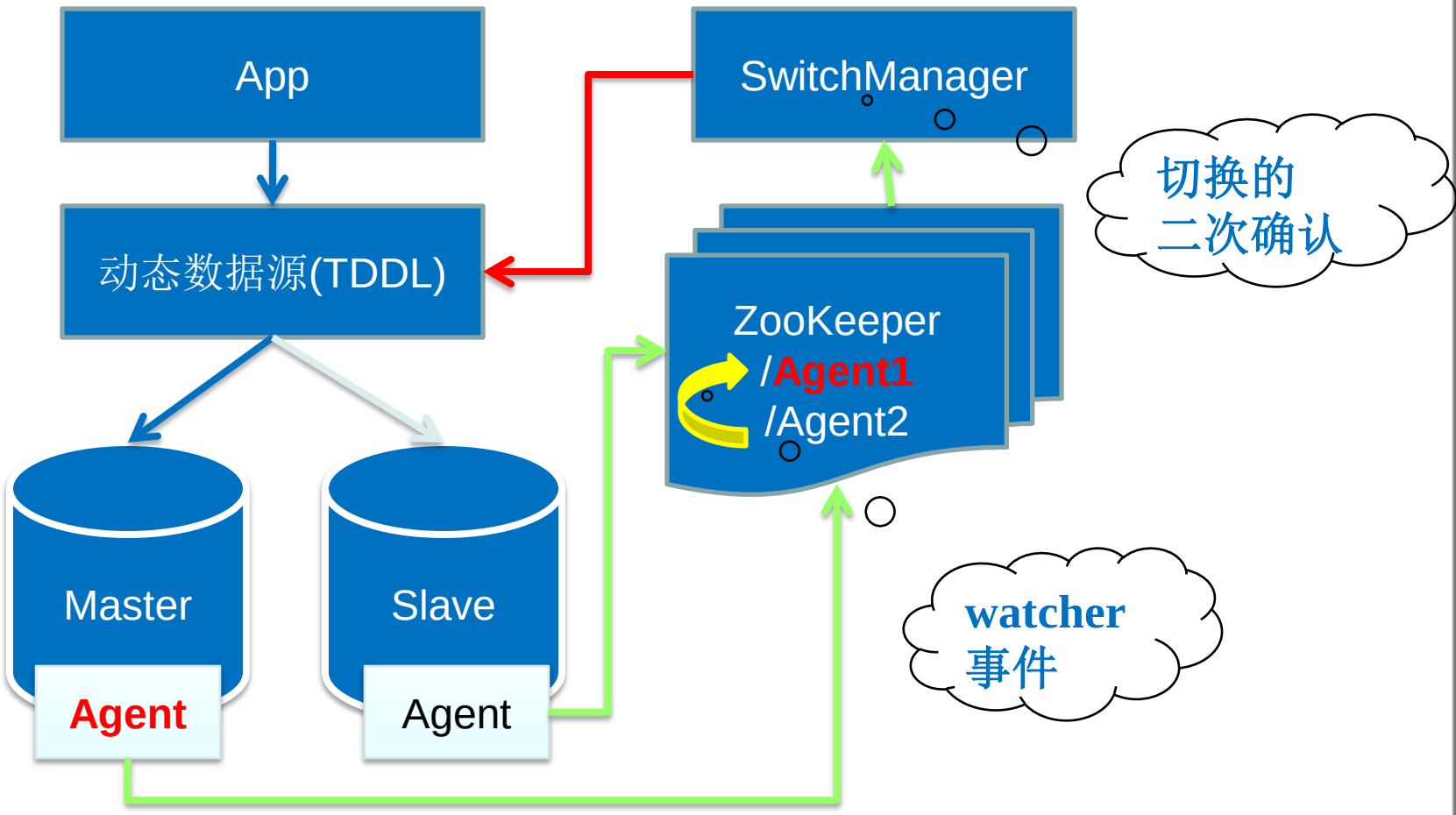
- 回滚Dead Master与新的主库一致(同步)
- 误删除的数据回滚 （**表级别、数据库级别**）
  - 主库做了一个delete、update的数据(row模式)
  - INSERT INTO test.ma(id) values (10);

```
/*!*/;
# at 944
# at 985
#120214 17:04:08 server id 54  end_log_pos 839  Table_map: `test`.`ma` mapped to number 33
#120214 17:04:08 server id 54  end_log_pos 883  Delete_rows: table id 33 flags: STMT_END_F
### DELETE FROM test.ma
### WHERE
###   @1=10
### DELETE FROM test.ma
### WHERE
###   @1=20
### DELETE FROM test.ma
### WHERE
###   @1=30
```

大纲

- MySQL高可用的难题
- TMHA的整体设计
- TMHA如何实现异常切换
- TMHA如何保证数据一致性
- TMHA如何实现自动切换
- TMHA如何解决主备库延迟
- 总结

自动切换(配置白名单即可)



1. OS、ping、mysql ping、mysql 读写自动判断即可
2. 配置白名单：
   在白名单里面的列表都可以自动切换（这一块是在SwitchManager里面控制）

大纲

- MySQL高可用的难题
- TMHA的整体设计
- TMHA如何实现异常切换
- TMHA如何保证数据一致性
- TMHA如何实现自动切换
- TMHA如何解决主备库延迟
- 总结

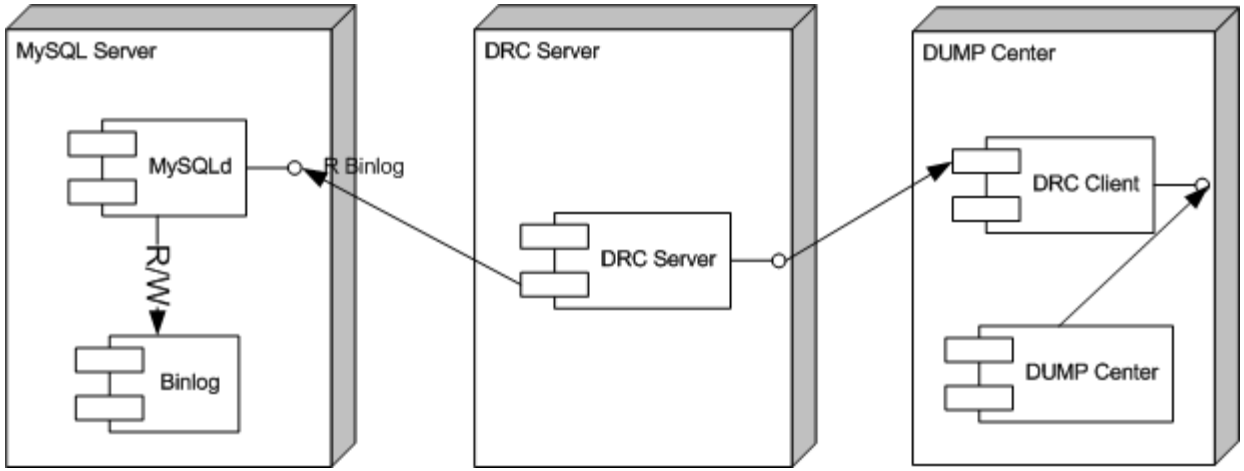
数据复制中心(DRC)

强一致

低延迟

高可用

- 架构图
  - 多线程写入目的端mysql等
  - 支持事务、dump给商品搜索



URL: http://www.taobaodba.com/html/772_data_replication_center.html

大纲

- MySQL高可用的难题
- TMHA的整体设计
- TMHA如何实现异常切换
- TMHA如何保证数据一致性
- TMHA如何实现自动切换
- TMHA如何解决主备库延迟
- 总结

总结

- 通过zookeeper实现配置的集中管理
- 数据一致性、read-only设置显得尤为重要
- 故障切换 + APP切换 + 人工/自动切换兼容

zk的监控改进

- 4-letter monitoring (mntr) / ganglia监控
- Taokeeper (中间件团队提供)

URL: http://zookeeper.apache.org/doc/r3.3.3/zookeeperJMX.html

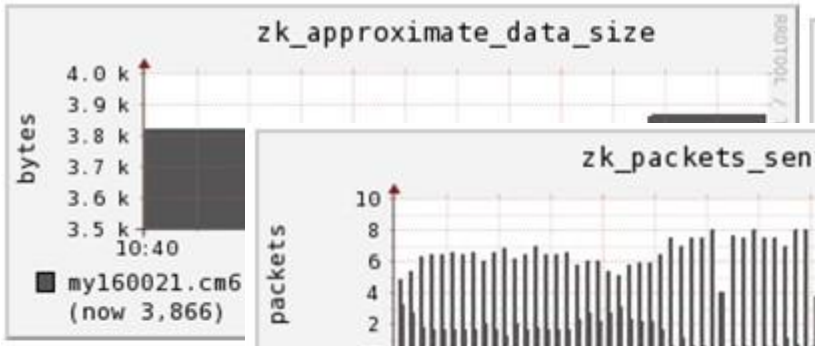
4-letter monitoring

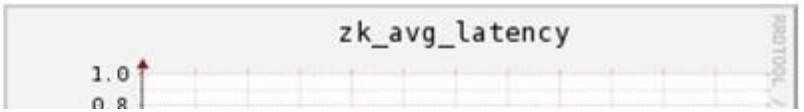
- 版本3.3.3需要添加patch-744方可(ant编译)
- 版本3.4自动支持(另外，3.4引入observer)

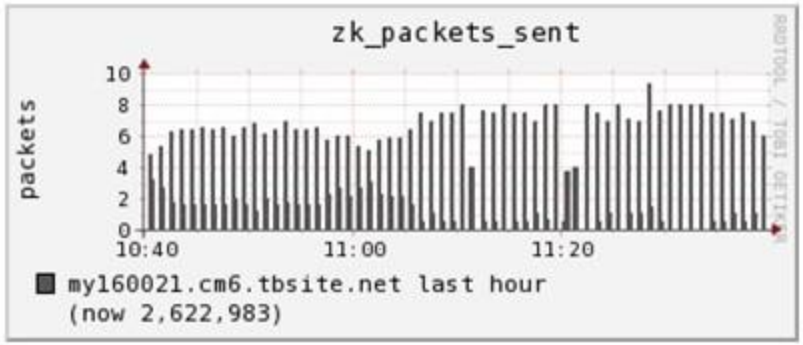
```
[mugong.zjq@v031061 bin]$  echo 'mntr' | nc localhost 2182
zk_version      3.3.3--1, built on 05/07/2011 12:31 GMT
zk_avg_latency  0
zk_max_latency  23
zk_min_latency  0
zk_packets_received     684721
zk_packets_sent 684720
zk_outstanding_requests 0
zk_server_state leader
zk_znode_count  8
zk_watch_count  1
zk_ephemerals_count     3
zk_approximate_data_size        157
zk_open_file_descriptor_count   26
zk_max_file_descriptor_count    1024
zk_followers    2
zk_synced_followers     2
zk_pending_syncs        0
[mugong.zjq@v031061 bin]$
```

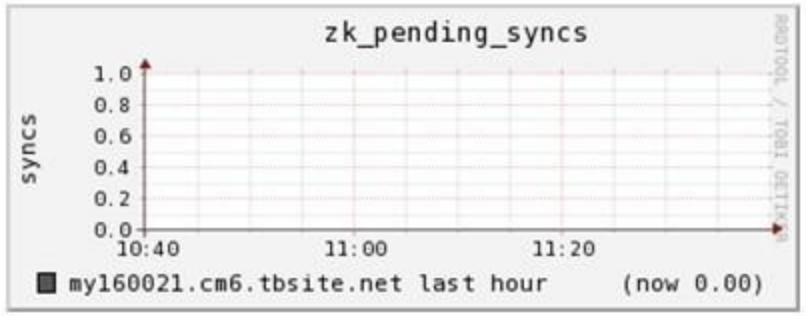
```
[mugong.zjq@v031061 bin]$  echo 'mntr' | nc localhost 2183
zk_version      3.3.3--1, built on 05/07/2011 12:31 GMT
zk_avg_latency  0
zk_max_latency  26
zk_min_latency  0
zk_packets_received     311984
zk_packets_sent 311983
zk_outstanding_requests 0
zk_server_state follower
zk_znode_count  8
zk_watch_count  1
zk_ephemerals_count     3
zk_approximate_data_size        157
zk_open_file_descriptor_count   23
zk_max_file_descriptor_count    1024
```

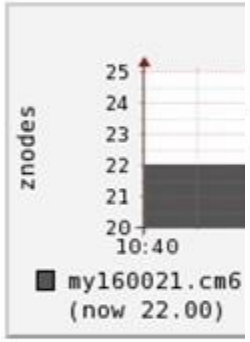
Ganglia监控

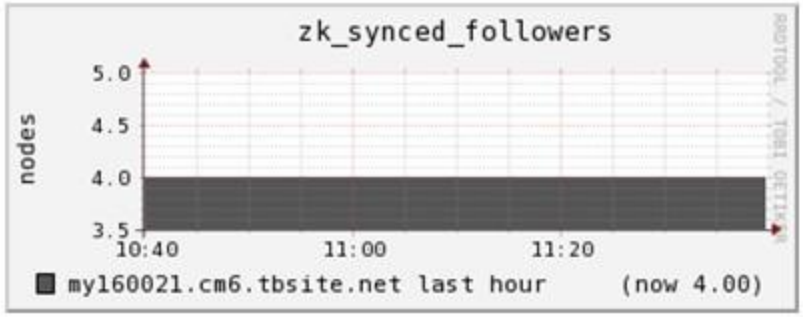

zk_approximate_data_size
bytes
4.0 k
3.9 k
3.8 k
3.7 k
3.6 k
3.5 k
10:40
my160021.cm6
(now 3,866)

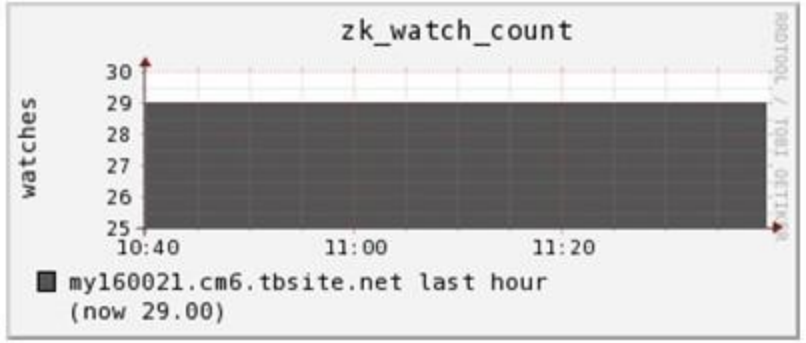

zk_avg_latency
1.0

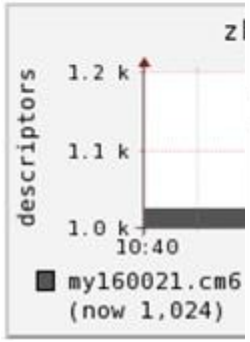

zk_packets_sent
packets
10
8
6
4
2
0
10:40
11:00
11:20
my160021.cm6.tbsite.net last hour
(now 2,622,983)

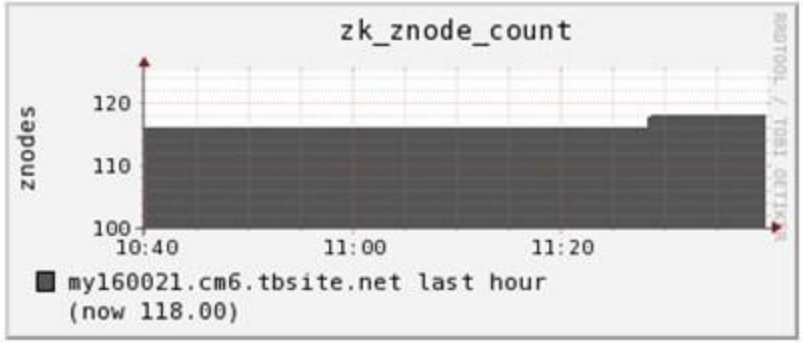

zk_pending_syncs
syncs
1.0
0.8
0.6
0.4
0.2
0.0
10:40
11:00
11:20
my160021.cm6.tbsite.net last hour
(now 0.00)


znodes
25
24
23
22
21
20
10:40
my160021.cm6
(now 22.00)


zk_synced_followers
nodes
5.0
4.5
4.0
3.5
10:40
11:00
11:20
my160021.cm6.tbsite.net last hour
(now 4.00)


zk_watch_count
watches
30
29
28
27
26
25
10:40
11:00
11:20
my160021.cm6.tbsite.net last hour
(now 29.00)


descriptors
1.2 k
1.1 k
1.0 k
10:40
my160021.cm6
(now 1,024)


zk_znode_count
znodes
120
110
100
10:40
11:00
11:20
my160021.cm6.tbsite.net last hour
(now 118.00)

Taokeeper监控

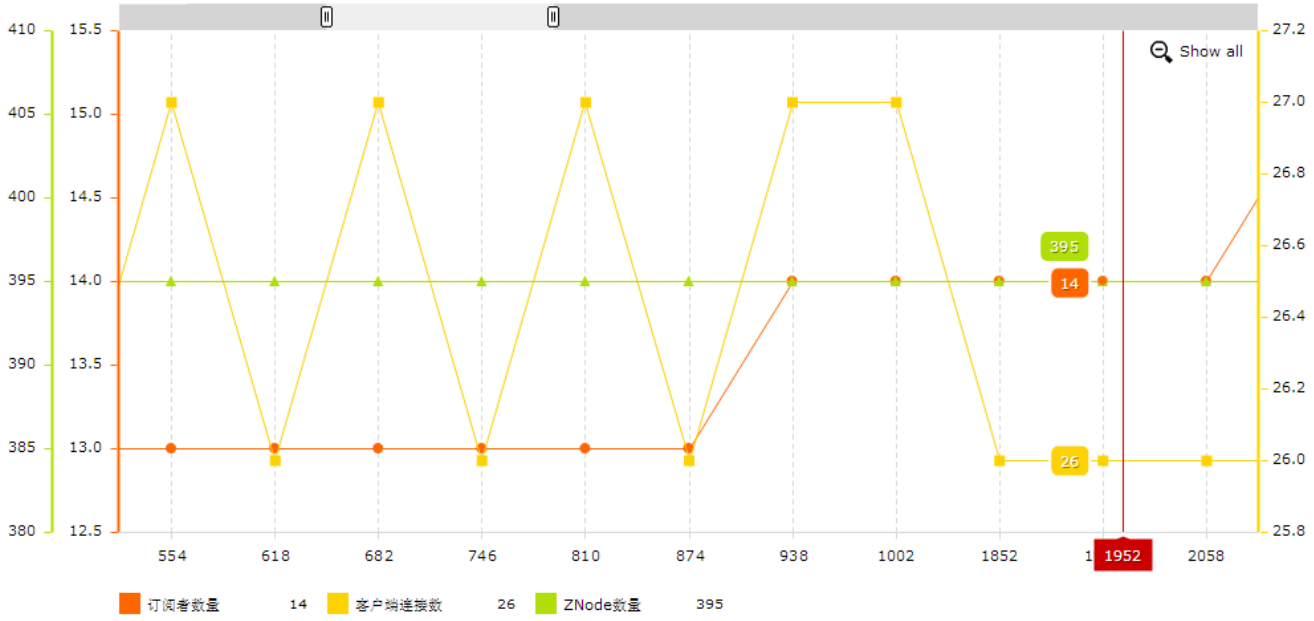
- Monitor
  - 集群配置
  - 集群监控
  - 机器监控
  - 报警设置
- Admin
  - 报警开关
  - 系统设置

ZooKeeper集群状态 更新时间：2012-12-01 00:51:03 加入监控

mysql_Auto_Transfer zk monitoring of mysql auto transfer

Node IP	Role	连接数	Watch数	Watched /Total Path	数据里 Sent/Received	状态	节点自检状态	查看趋势
72	F	37	14	12/396	17135123/17099785		OK	
2	F	42	20	20/396	21916825/21876096		OK	
1	L	32	14	14/396	13852568/13851712		OK	
5	F	21	11	11/396	9655666/9660691		OK	
							Checking	

chart by amcharts.com



Q&A

微博：淘穆公
http://www.weibo.com/**suinking**
Email： mugong.zjq@taobao.com

REF:
http://www.slideshare.net/suinking/v20-9043338
http://www.suinking.net/?p=32