淘宝网
Taobao.com

DTCC2013

阿里数据库关键技术

张瑞 @ Taobao

个人介绍

- 张瑞，HelloDBA，Oracle ACE
- 2005年加入阿里数据库团队
- Oracle DBA -> MySQL DBA -> DA
- 2012年，参与翻译《Expert Oracle Exadata》
- 个人博客：Hello Database(hellodb.net)
- AskHelloDBA技术论坛
- 新浪微博：hellodba

数据库关键技术

- 系统软硬件概况
- 分布式数据库访问层
- 数据库自动扩容工具
- 淘宝MySQL高可用
- 阿里MySQL工具集
- 应用和系统优化

双11数据

- 全天成交额：191亿
- 全天订单数：1亿笔
- 数据库峰值数据：
  - 单机QPS：40000
  - 单机TPS：10000
  - 单机逻辑读：5000000
  - 单机物理读：8000

数据库软硬件

- 硬件
  - PC Server
  - Intel E5645
  - 48G或96G Memory
  - 12 SAS或8 SSD + 2 SAS
  - PCI-E Flash卡
- 数据库
  - MySQL 5.5

数据库存储方案

- 存储方案
  - Flashcache
  - Flash卡+SAS
  - SSD+SAS
  - SAS
- 选择标准
  - 数据大小
  - 性能要求
  - 应用模型

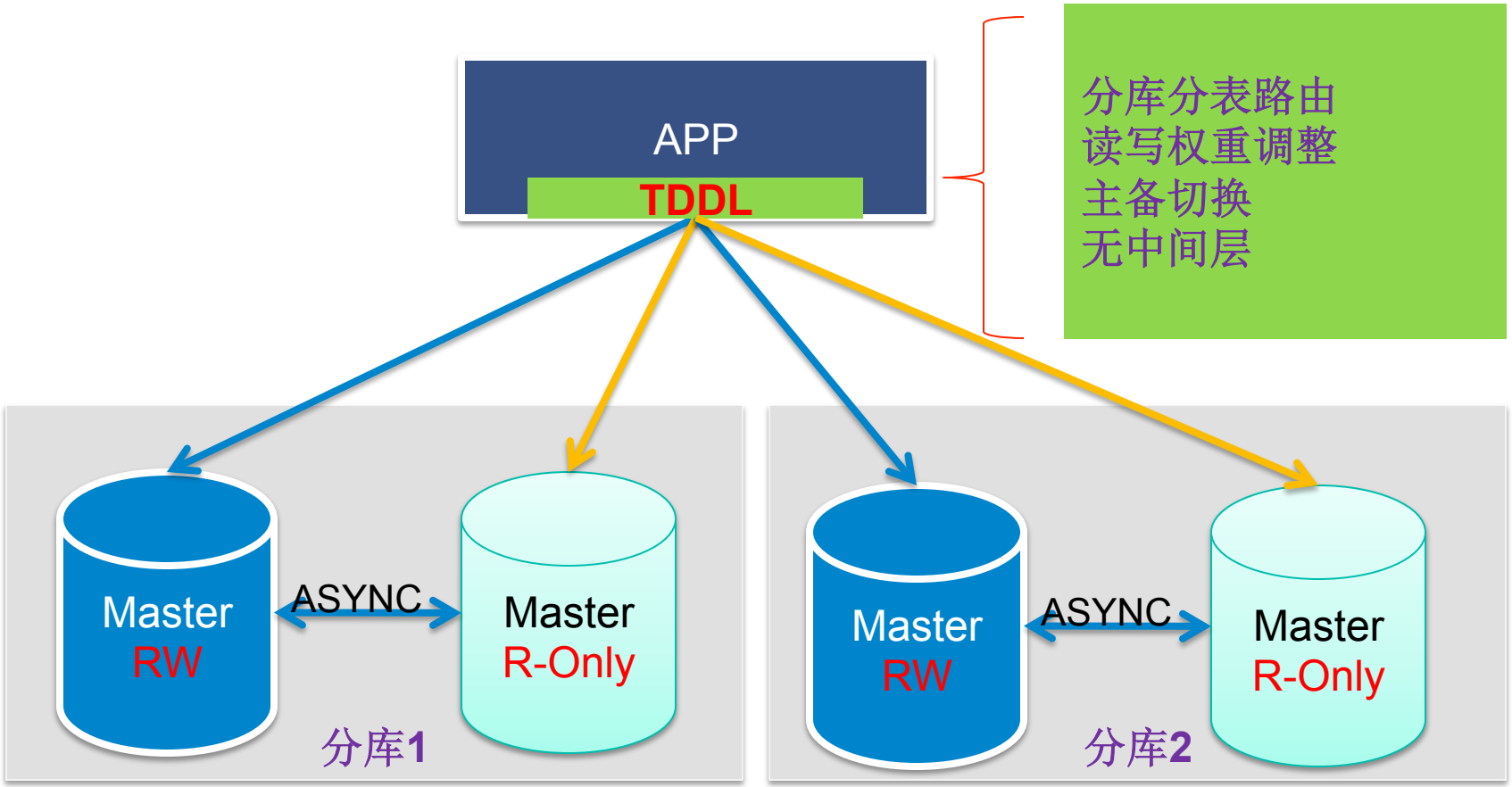
数据库架构

- 可扩展架构
  - 分库分表
  - 读写分离
- 高可用架构
  - M-M
  - M-M-S
- 实例管理
  - 单机多实例
  - 资源隔离

MySQL 5.5 参数

- innodb_flush_log_at_trx_commit=1
- innodb_thread_concurrency=64
- innodb_adaptive_hash_index_partitions=8
- innodb_buffer_pool_instances=8
- innodb_flush_method=O_DIRECT
- innodb_adaptive_flushing=1
- innodb_adaptive_flushing_method=keep_average
- innodb_stats_on_metadata=0
- innodb_use_native_aio=1
- innodb_flush_neighbor_pages=0
- innodb_change_buffering=inserts
- transaction-isolation=READ-COMMITTED
- Innodb_old_blocks_time=1000
- sync_binlog=1
- binlog-format=rows

分布式数据中间层(TDDL)

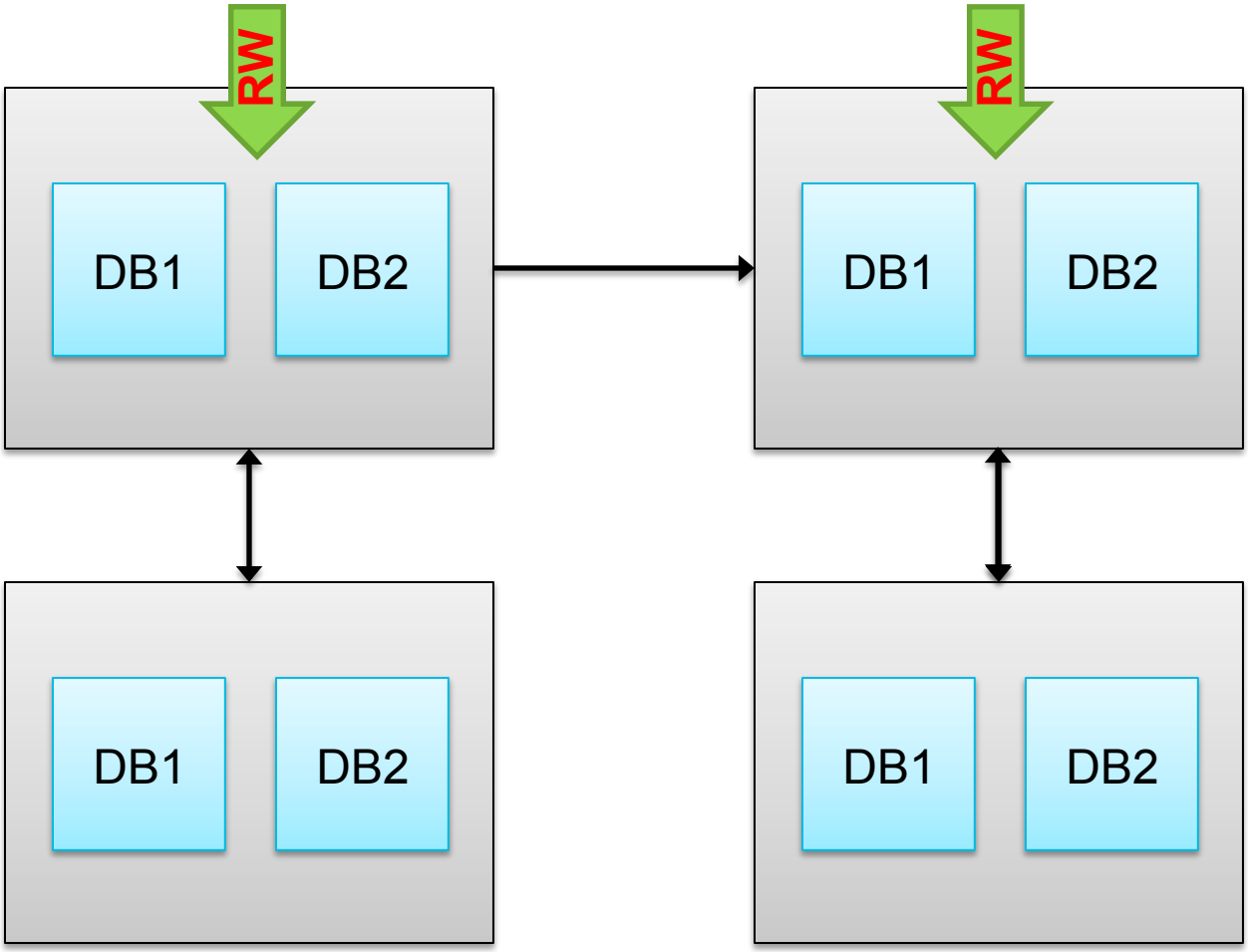


1. Master和Master-Readonly的mysql部署在不同机房
2. 异步复制，有数据延迟
3. 分库分表

数据库扩容方案

- 集群扩容
  - 数据库水平扩展，2主2备->4主4备
  - 针对TPS容量不足的核心数据库
  - 扩容后缩减比较困难
- 机器升级
  - 升级为SSD，提升IO性能
  - 内存扩容，提升buffer命中率
- 增加备库
  - 增加MySQL备库，应用读写分离
  - 针对QPS容量不足的场景
  - 扩容和缩减很方便

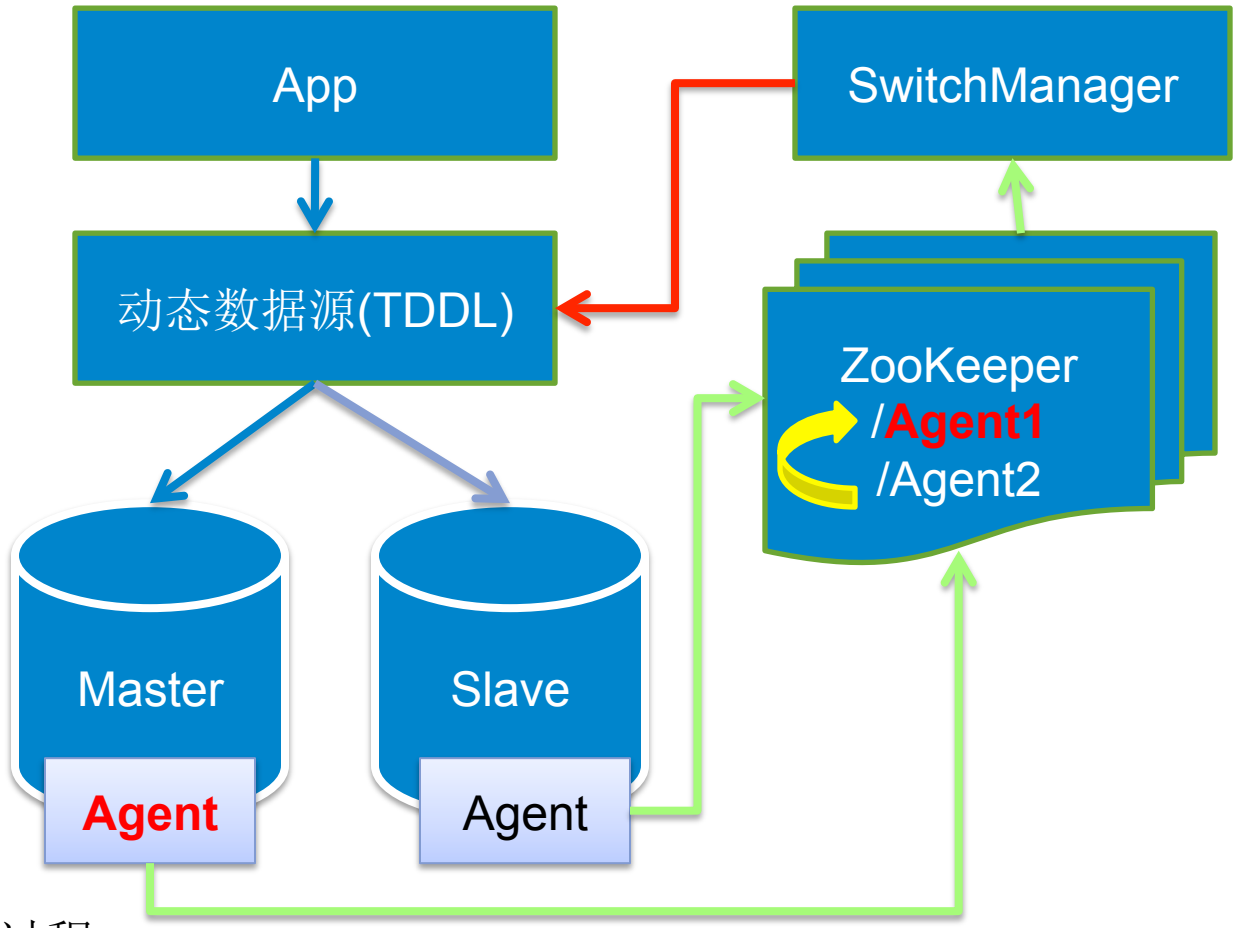
数据库自动扩容



1. 搭建备库
2. 主库停写
3. 检查主备一致
4. 停止新旧复制
5. 修改复制关系
6. 删除冗余DB
7. 推送分库规则
8. 打开主库读写

DBFree是数据库自动扩容/缩减工具

MySQL高可用(TMHA)



异常切换过程：

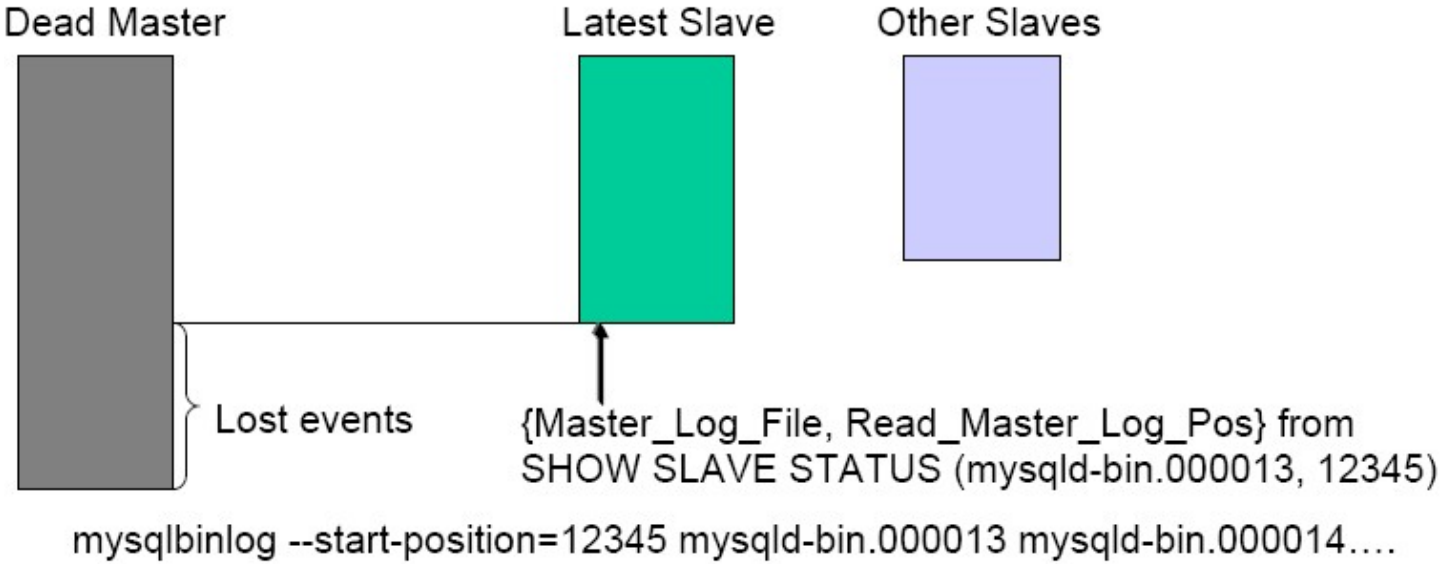
1. Master宕机，zookeeper的agent1结点消失
2. Agent2得知watcher事件，记录异常，创建异常结点
3. SwitchManager获取最新的异常结点，再次确认状态
4. 主备库切换：推送TDDL配置，将新主库置为可写

MySQL高可用(TMHA)

- 切换类型
  - 正常切换
  - 强制切换
  - 批量切换
- 部署方式
  - MySQL主备库部署在不同机房
  - Zookeeper部署在三个机房
- 优点
  - 多机房部署可实现IDC容灾

TMHA如何保证数据一致性

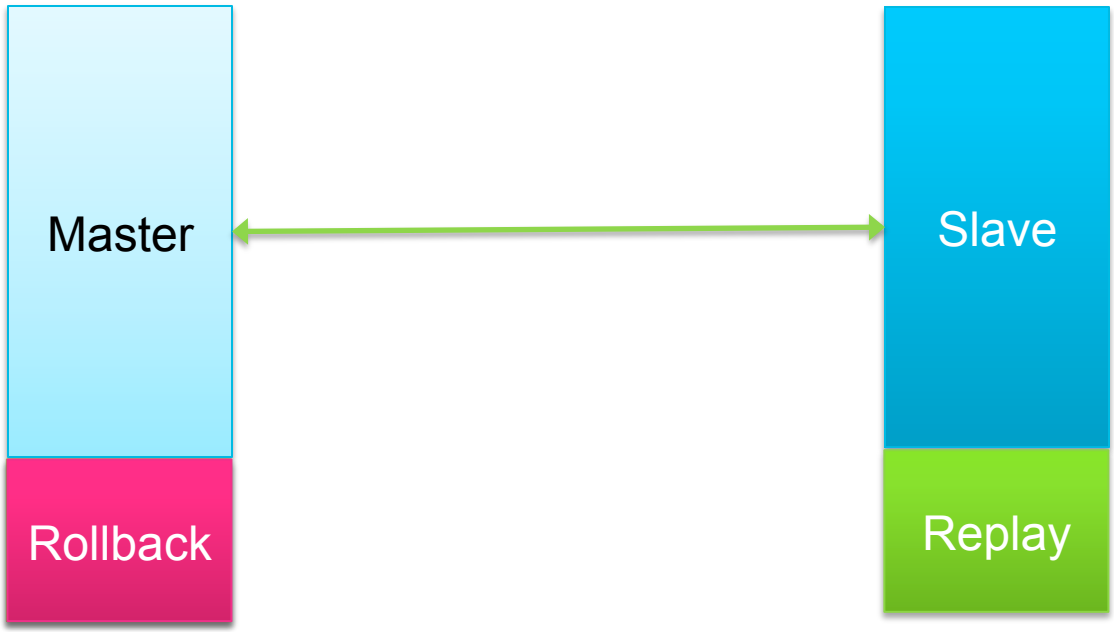
- 传统思路
  - 共享存储
  - 半同步
- 无共享存储
  - innodb_flush_log_at_trx_commit=1
  - sync_binlog=1
  - innodb_support_xa=true



如何保证数据一致性

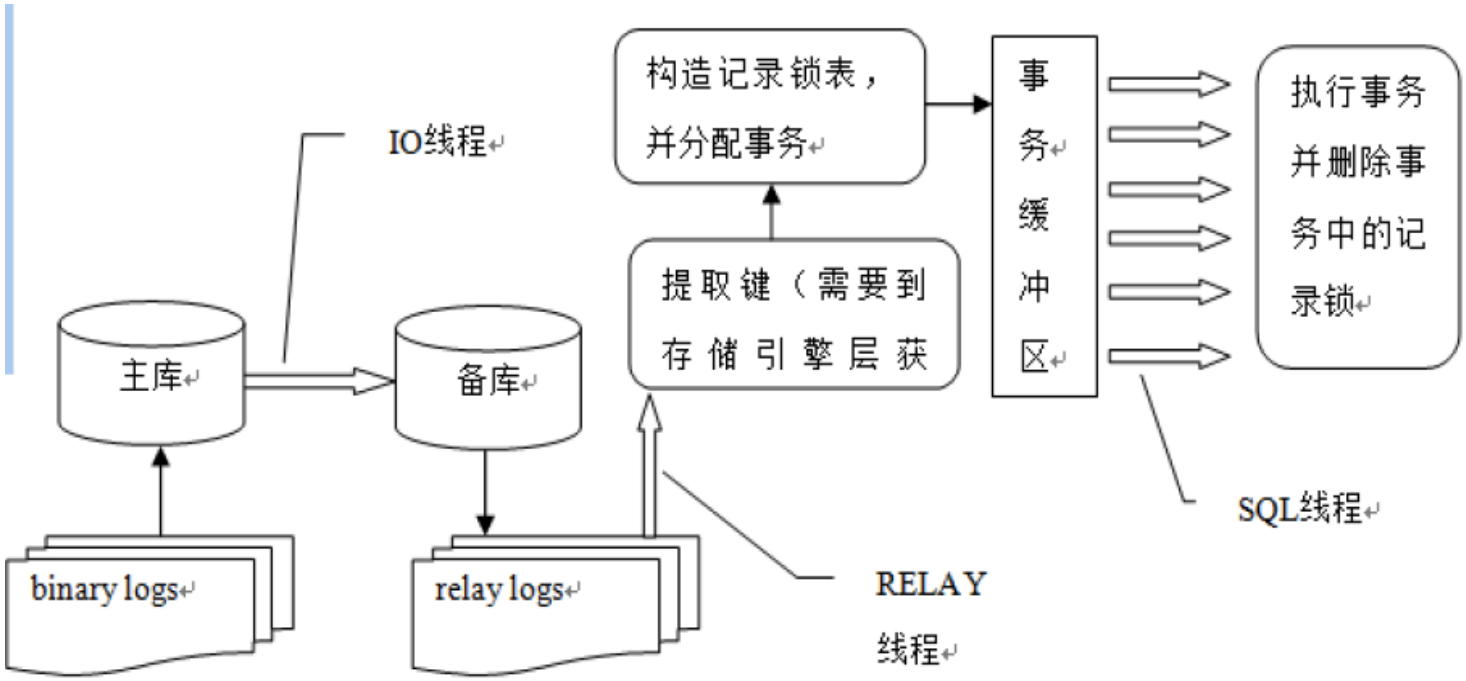
- Master宕机后，三个选择：
  1. Slave立即提供服务，存在数据不一致风险
  2. Slave不提供服务，等待master恢复，保证数据一致
  3. Slave提供部分服务(比如只能新建，不许修改)，等待master恢复后，保证数据一致
- TMHA的处理策略：
  1. Slave立即提供服务
  2. Slave(旧) -> Master(新)
  3. Master(旧) Rollback
  4. Master(旧) -> Slave(新)
  5. Master(新) Replay

Rollback & Replay



- Rollback
  - Master回滚，保证与Slave一致
  - 重新恢复主备复制关系
- Replay
  - Slave重放，减少数据丢失
  - 冲突检测机制

- MySQL并行复制
  - 三种并发模式：事务，表或库
  - 兼容原生复制，可随时切换
  - 语法：start slave multi_sql_thread



show processlist可看到多个复制线程

```
root@(none) 09:43:35>show processlist;
+-------+------------+----------+------+---------+--------+-----------------------------------------+------------------+
+--------------+-----------+
| Id    | User       | Host     | db   | Command | Time   | State                                   | Info             |
| Rows_examined | Rows_read |
+-------+------------+----------+------+---------+--------+-----------------------------------------+------------------+
+--------------+-----------+
|    30 | root       | localhost | NULL | Sleep   |      1 |                                         | NULL             |
|           350 |       351 |
|    72 | system user |          | NULL | Connect | 136635 | Queueing master event to the relay log | NULL             |
|             0 |         1 |
|    73 | system user |          | NULL | Connect |     18 | Waiting for the active trx in trx list | NULL             |
|             0 |         1 |
|    74 | system user |          | NULL | Connect |     18 | Waiting for the active trx in trx list | NULL             |
|             0 |         1 |
|    75 | system user |          | NULL | Connect |     18 | Waiting for the active trx in trx list | NULL             |
|             0 |         1 |
|    76 | system user |          | NULL | Connect |     18 | Waiting for the active trx in trx list | NULL             |
|             0 |         1 |
|    77 | system user |          | NULL | Connect |     18 | adding records to trx lock table       | NULL             |
|             0 |         1 |
| 12119 | root       | localhost | NULL | Query   |      0 | NULL                                    | show processlist |
|             0 |         1 |
+-------+------------+----------+------+---------+--------+-----------------------------------------+------------------+
+--------------+-----------+
8 rows in set (0.02 sec)
```

阿里工具集

- orzdba/orzcluster：MySQL实时性能监控工具
- orztop：MySQL实时SQL监控工具
- rollback：MySQL binlog回滚工具
- relay-fetch：slave预读，提升复制性能
- slave-error-handle：复制错误处理工具
- tbsql：数据库日常管理工具集
- tbsync：主备数据对比工具
- myddl：在线表结构修改

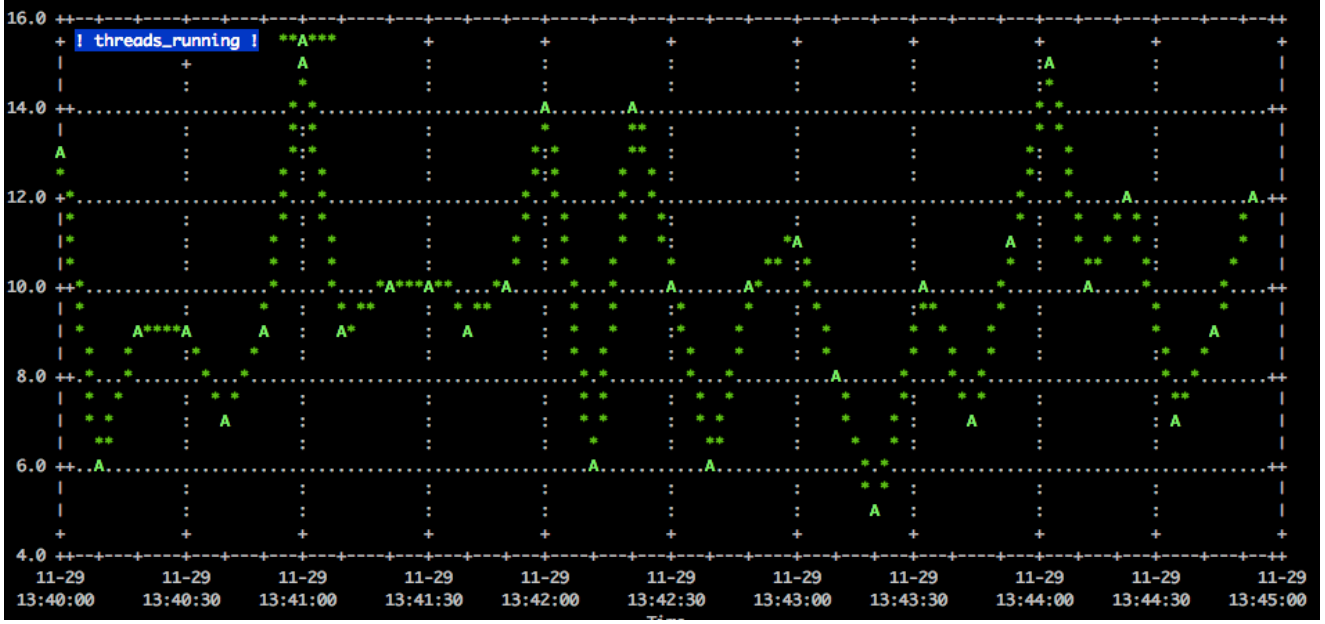
……

阿里工具集

```
binlog_format[ROW] max_binlog_cache_size[8G] max_binlog_size[500M]
max_connect_errors[50000] max_connections[6000] max_user_connections[5900]
open_files_limit[65535] sync_binlog[0] table_definition_cache[2048]
table_open_cache[2048] thread_cache_size[256]

innodb_adaptive_flushing[ON] innodb_adaptive_hash_index[ON] innodb_buffer_pool_size[72G]
innodb_file_per_table[ON] innodb_flush_log_at_trx_commit[2] innodb_flush_method[O_DIRECT]
innodb_io_capacity[1200] innodb_lock_wait_timeout[100] innodb_log_buffer_size[200M]
innodb_log_file_size[1.26953125G] innodb_log_files_in_group[3] innodb_max_dirty_pages_pct[50]
innodb_open_files[65535] innodb_read_io_threads[8] innodb_thread_concurrency[32]
innodb_write_io_threads[8]
```

--------	-----load-avg----			---cpu-usage---				---swap---		-QPS- -TPS-				-Hit%-	------threads------			
time	1m	5m	15m	usr	sys	idl	iow	si	so	ins	upd	del	sel	iud	lor	hit	run	con	cre	cac
13:52:06	1.95	2.20	2.34	5	1	94	0	0	0	0	0	0	0	0	0	100.00	0	0	0	0
13:52:07	1.95	2.19	2.34	7	2	90	1	0	0	96	439	0	6106	535	288032	99.37	11	2476	0	85
13:52:08	1.95	2.19	2.34	7	2	90	1	0	0	102	484	0	6432	586	252523	99.10	14	2476	0	85
13:52:09	1.95	2.19	2.34	7	2	91	1	0	0	85	439	0	5494	524	294271	99.21	6	2476	0	85
13:52:10	1.95	2.19	2.34	6	2	92	1	0	0	55	399	32	5475	486	201729	99.07	9	2476	0	85
13:52:11	1.95	2.19	2.34	6	2	92	0	0	0	64	337	0	5096	401	202295	99.10	10	2476	0	85
13:52:12	2.03	2.20	2.34	7	3	90	1	0	0	116	456	0	6282	572	212684	99.12	7	2476	0	85
13:52:13	2.03	2.20	2.34	7	2	91	1	0	0	101	417	0	5791	518	218048	99.06	7	2476	0	85



应用优化

- 减库存，抢红包场景
  - 大量并发更新导致行锁等待严重
  - 触发MySQL死锁检测，CPU耗尽
  - thread running剧烈波动，RT上升
- MySQL补丁：
  - 关闭死锁检测
  - 合并更新
- 应用优化：
  - 库存或红包拆分
  - 更新cache，异步写DB

系统优化

- 无处不在的cache
  - 降低DB的读压力
  - Cache失效怎么办
- 系统解耦
  - 减少系统依赖
  - 保护核心应用
- 系统保护
  - 降级开关
  - 自动限流

路走对了，就不怕远！

我的联系方式：
微博：hellodba
博客：www.hellodb.net
邮件：freezr@gmail.com

谢谢