

Mysql快速讲义

峰云就她了

xiaorui.cc



about me

Name: 峰云就她了

Blog: xiaorui.cc

Github: github.com/rfyiamcool

Desc: 喜欢研究高性能服务端、数据库相关



LIST

01

Mysql Basic

02

Mysql Ops

03

Mysql Optimize

04

Mysql集群



基本选择

版本

* Mysql \geq 5.7

* Percona # recommend

* Mariadb

* more ...

秒杀压测

测试: update goods set stock =stock -1 where id = 1 and stock >3

并发连接	mysql 5.7	percona5.7	mysql (关闭死锁)	percona (关闭死锁)
256	2000	4000	6000	4200
512	500	4300	2500	4350
1024	30	5000	1500	5200

引擎的选择

- * Myisam # 淘汰的玩意
- * Innodb # 优秀全能手
- * Tokudb # 高压缩率
- * Myrocks # 基于lsm的rocksdb

表结构设计

- * `innodb`一定要有主键
- * 主键最好整型, 并单调递增
- * 将`text/blob`大对象独立存放
- * 要注意字符集/ 校验集的一致性, 避免类型隐式转换
- * 尽量不要使用外键

表使用规范

- * 使用Not null, default ``
- * 少用limit m, n取范围
- * 多用limit, 减少检索及输出
- * 避免 select *
- * 避免使用 存储过程

表使用规范

- * 表的索引不要太多
- * 表的字段不要太多
- * 单表行数在1000w

索引基本规则

- * $\text{idx}(a, b, c)$ HIT where $a = x$ and $b = x$
- * $\text{idx}(a, b, c)$ HIT where $a > x$
- * $\text{idx}(a, b, c)$ Not HIT where $b > x$
- * $\text{idx}(a, b, c)$ Not HIT where $a > x$ and $b = x$
- * $\text{idx}(a, b, c)$ Not HIT where $a = x$ and $c = x$

索引基本规则

- * `idx(sex)`

- * 没必要建立索引

- * 显示 和 隐式 `join` 性能与结果一致

- * `from a,b` vs `from a inner join b on`

索引高级规则

- * $\text{idx}(a, b, c)$ HIT where $a = x$ order by b
- * $\text{idx}(a, b, c)$ HIT where $a > x$ order by a
- * $\text{idx}(a, b, c)$ HIT where $a = x$ and $b > x$ order by a
- * $\text{idx}(a, b, c)$ Not HIT where $a > x$ order by b
- * $\text{idx}(a, b, c)$ HIT where $a = x$ group by a, b
- * $\text{idx}(a, b, c)$ Not HIT where $a = x$ group by b

联表索引



驱动表



笛卡尔积



nested loop join

```
for each row in t1 matching range {  
  for each row in t2 matching refer key {  
    for each row in t3 {  
      if row satisfies join conditions,  
        send to client  
    }  
  }  
}
```

联表查询

- * 简单说, 小表驱动大表
- * 右表的条件列一定要加上索引
- * 显示 和 隐式 join 性能与结果一致
 - * from a, b vs from a inner join b on
- * Inner自动选择驱动表, left join 选择左面表.

关键

- * 多用Explain
- * 多看慢查询
- * 优化器会随着数据量变大发生变化

explain

- * using index;
- * using where;
- * using where; using index;
- * using filesort;
- * using temporary;

High

Low



排查问题

事务锁

通过三表信息能快速发现哪些事务在阻塞其他事务



INFORMATION_SCHEMA.INNODB_TRX



INNODB_LOCKS



INNODB_LOCK_WAITS

慢查询

开启定义慢查询条件及开关

- * `slow_query_log` # 开关
- * `slow_query_log_file` # 日志路径
- * `long_query_time` # 超时时间

管理连接

* show processlist; # 正在进行的sql语句 .

* Sleep

* Sending data

* Waiting for tables

* more...

* kill id;

* 杀掉某任务



Ops Mysql

修改表结构



阻塞



alter table xxxx



非阻塞



pecona/ pt-online-schema-change



触发器方案



github/ gh-ost



模拟主从协议

percona toolkit

- * pt-heartbeat
- * pt-table-checksum
- * pt-table-sync
- * pt-archiver
- * pt-kill

热备份

- * Mysqldump
 - * 非一致性备份
- * Mysqlpump
 - * 基于表并发
- * Percona XtraBackup

binlog2sql

1. binlog 日志格式为 row
2. 找到相关的sql pos点
3. 导出回溯sql语句
4. 导入重写修改的数据



后悔药



Optimize Mysql

innodb

- * innodb_buffer_pool_size
- * innodb_log_file_size
- * innodb_log_buffer_size
- * innodb_flush_log_at_trx_commit
- * innodb_file_per_table
- * innodb_buffer_pool_instances
- * more

innodb

- * autocommit
- * innodb_deadlock_detect
- * innodb_lock_wait_timeout

FB: 可以这么调优lock_deadlock ...

Mysql: 方案不靠谱

FB: 我写完了, 已提pr了

Mysql: 已经merge了

sql

- * 连接池复用
- * 批量更新减少rtt消耗
- * 避免大数据返回
- * 适当的压缩数据
- * 多用乐观锁
- * more ...

cmd

- * `iostat -x 1`
- * `perf`
- * `dstat`
- * `free -m`
- * `mpstat -p ALL`
- * `top`

io

最重要 !!!

cpu

再提高



硬件



升级 SSD



升级 大内存



分布式



分库分表



中间件



system



文件系统xfs/zfs



软中断

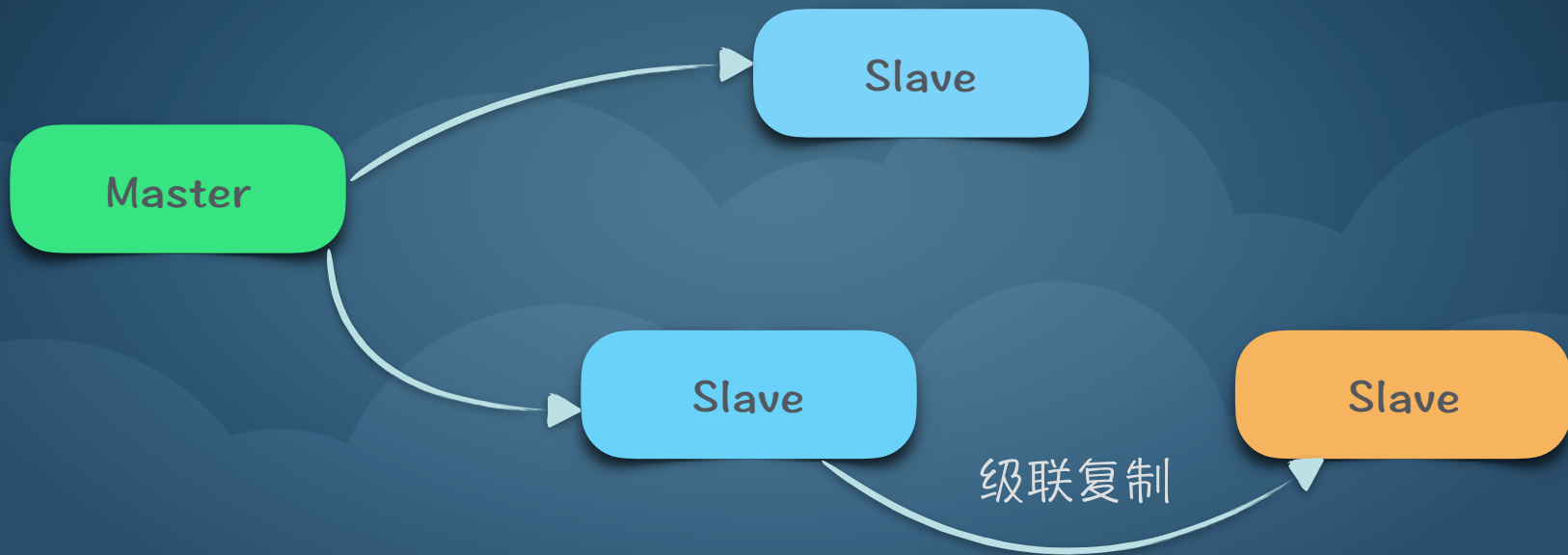


磁盘算法



mysql cluster

Master/Slave



HA



Slave convert new Master !!!

可靠 ?

- * slave replication
- * semi replication
- * semi replication +

主从切换的集群环境下:
不能保证消息的一致性

how ?

- * 阿里自研

- * 腾讯自研

- * 京东自研

- * 各大云厂商自研

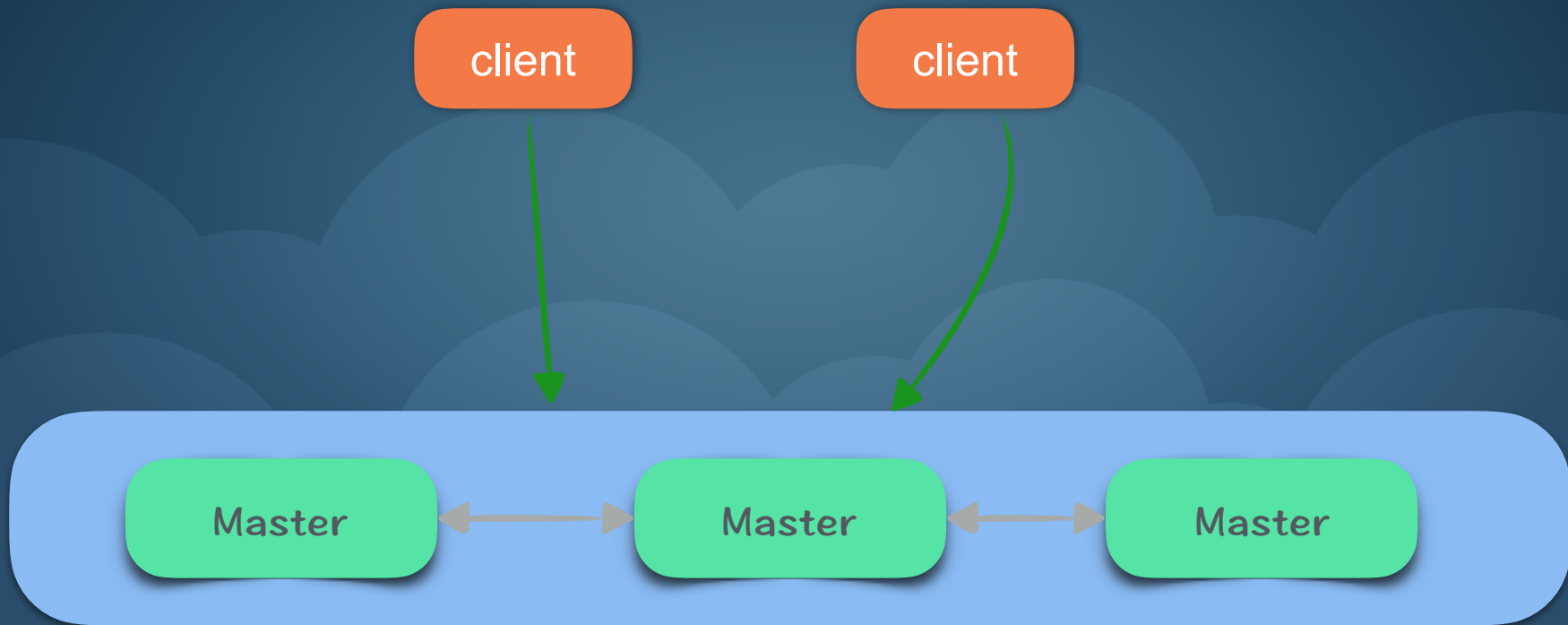
- * more

通过paxos, raft来保证消息的一致性.

HA

- * keepalived multi master repl
- * heartbeat drdb
- * mysql galera
- * mha # recommend
- * mysql group replication # recommend

MGR



中间件

- * Mycat # recommend
- * Kingshard
- * DBProxy
- * Cobar
- * More

Q & A

